UNIVERSITY OF EDUCATION, WINNNEBA

A SURVEY OF ADVANCE HOMOMORPHIC ENCRYPTION SCHEMES USED TO SECURELY STORE DATA IN CLOUD

CARVEN ABALENA

A Dissertation in the Department of Information Technology Education, Faculty of Technical Education, submitted to the School of Graduates Studies in Partial Fulfilment of the requirements for the award of the degree of Master of Science (Information Technology Education) in the University of Education, Winneba.

MAY, 2020

DECLARATION

STUDENT'S DECLARATION

I, CARVEN ABALENA, declare that this dissertation, with the exception of quotations and references contained in published works which have all been identified and duly acknowledged, is entirely my own original work, and it has not been submitted, either in part or whole, for another degree elsewhere.

SIGNATURE…….....

Date……

SUPERVISOR'S DECLARATION

I hereby declare that the preparation and presentation of this work was supervised in accordance with the guidelines for supervision of dissertation as laid down by the University of Education, Winneba.

NAME OF SUPERVISOR: MR. WILLIAM ASIEDU

SIGNATURE: ……

DATE: …… …

DEDICATION

I wish to dedicate this research work to my lovely mum Achegebakem Augustina, and father Abalena Chawuru, all of blessed memory. I also dedicate it to my sweet heart and soul mate Helen Wegwi, my daughters Wewoke Faith and Ajegewe Redeemer for their decent support towards my education and success of this dissertation.

ACKNOWLEDGEMENT

Hardly can I thank all those who have assisted, guided and contributed in various ways directing the writing of this dissertation. However, my sincere and profound gratitude goes to Almighty God who kept me and granted me the wisdom, knowledge and strength to be able to finish this task. In fact, I actually appreciated his hand work. My sincere gratefulness also goes to my supervisor, Mr. William Asiedu for guiding me throughout this dissertation writing.

To my family, I appreciated and thanked you for your encouragements and advices you gave me. I thanked Almighty God for blessing me with such a lovely family.

To my colleague teachers of Awe Senior High/ Technical School especially those at ICT department I am grateful to you for your support and encouragement.

Finally, to my friends, I thanked you all for your time and support.

TABLE OF CONTENT

CONTENT **PAGE**

DECLARATION iii

DEDICATION iv

ACKNOWLEDGEMENT v

TABLE OF CONTENT vi

LIST OF TABLES x

LIST OF FIGURES xi

ABSTRACT xii

CHAPTER ONE: INTRODUCTION 1

1.1 Introduction 1

1.2 Background to the Study 2

1.3 Statement of the Problem 4

1.4 Aim and Objectives of the study 4

1.5 Research Questions 5

1.6 Significance of the Study 5

1.7 Limitation 6

1.8 Delimitation 6

1.9 Definition of Terms 7

1.10 List of Abbreviations 8

1.11 Organization of the Study 9

CHAPTER TWO: LITERATURE REVIEW 10

2.1 Introduction 10

2.2 Conceptual Framework 11

2.3 Encryption 12

2.3.1 Symmetric Encryption 13

2.3.2 Asymmetric Encryption 14

2.4 Cloud Computing models 14

2.4.1 Cloud computing service models 15

2.4.1.1 Infrastructure-as-a-service (IaaS) 15

2.4.1.2 Platform-as-a-Service (PaaS) 16

2.4.1.3 Software-as-a- Service (SaaS) 16

2.4.2 Cloud computing deployment models 17

2.4.2.1 Private Cloud 17

2.4.2.2 Public cloud 17

2.4.2.3 Hybrid Cloud 18

2.4.2.4 Community cloud 18

2.5 Security of Cloud Computing 18

2.5.1 Cloud Computing Security Framework 20

2.5.1.1 Firewall 20

2.5.1.2 Security Measures of SaaS 21

2.5.1.3 Security Measures of PaaS Layer 22

2.5.1.4 Security Measures of IaaS Layer 22

2.6 Homomorphic Encryption for Data security and privacy in Cloud 23

2.6.1 Categories of Homomorphic Encryption 24

2.6.1.1 Partially Homomorphic Encryption 25

2.6.1.1.1 The RSA Encryption Scheme 25
2.6.1.1.1.1 Time Complexity of RSA Cryptosystem 27
2.6.1.1.2 Elgamal Encryption Scheme 28
2.6.1.1.2.1The time complexity of Elgamal Cryptosystem 28
2.6.1.2 Fully Homomorphic Encryption 29
2.6.1.2.1 DGHV Encryption Scheme 31
2.6.1.2.1.1Time Complexity of DGHV scheme 32
2.6.1.2.2 SDC Encryption Scheme 33
2.6.1.2.2.1Time Complexity of SDC scheme 34
2.6.1.2.3 SAM Encryption Scheme 35
2.6.1.2.3.1Time Complexity of SAM scheme 38
2.6.1.2.4 SA Encryption Scheme 38
2.6.1.2.4.1 Time Complexity of SA scheme 40
2.6.1.2.5 FHE in Cloud Security 41
2.6.2 Execution time of the homomorphic encryption schemes 43
2.6.3 Security of the Homomorphic Encryption Schemes 45
CHAPTER THREE: METHODOLOGY 47
3.1 Introduction 47
3.2 Research Process 47
3.4 Rationale/Justification for Systematic Mapping Studies 50
3.5 Search Strategy 52
3.5.1. Inclusion-Exclusion Criteria 52
3.5.1.1 Inclusion 53

3.5.1.2 Exclusion 53
3.5.2 Execution Process 53
3.5.3 Conducting the Search 54
3.6 Data Analysis and Classification 57
3.7 Threats to Validity/Limitations 58
3.8 Conclusion 60
CHAPTER FOUR: DISCUSSION OF FINDINGS 61
4.1 Introduction 61
4.2 Discussion of Parameters 61
4.3 Research Question one (1): What are the current advance homomorphic encryption schemes available in the cloud? 62
4.4 Research Question two (2): Which advance Homomorphic Encryption Schemes is the best for storing data securely in cloud? 64
CHAPTER FIVE: SUMMARY, CONCLUSION, RECOMMENDATION AND FUTURE WORK 68
5.1 Introduction 68
5.2. Summary of Findings 68
5.3 Conclusion 69
5.4. Recommendations 70
5.5 Future Work 71
REFERENCES 73

LIST OF TABLES

TABLE **PAGE**

1: The execution time of homomorphic encryption schemes, SAM, SDC and DGHV 43

2: Execution Time of SA, Egamal and RSA Schemes..... 44

3: Description of the steps in Review 55

4: The search results and selected articles from each digital library..... 56

5: Research categories and number of articles..... 58

6: Size of message, execution time, time complexity and security of homomorphic encryption schemes..... 61

LIST OF FIGURES

FIGURE **PAGE**

1: Structural design of cloud storage 12

2: Cloud Computing Security Framework..... 20

3: Categories of Homomorphic Encryption..... 25

4: Application of total homomorphic encryption to secure the cloud 42

5: The Data Security Scheme for Cloud Computing..... 42

6: Flow chart of the research process 48

7: Systematic Mapping Process..... 49

ABSTRACT

Work has been conducted to determine the advanced homomorphic encryption scheme that is ideally suited to securely store data in the cloud. Some cloud service providers share information with third parties in order to provide effective cloud services. The third party can access the client's private data and modify the information to make it beneficial to the client. In cloud computing, the client's data is put in the cloud, and any computation of the data stored by the client is implemented in the cloud. Security is the biggest problem about cloud storage as a service provider can access, deliberately alter or even delete stored data. This concern made the researcher to find out the best advance Homomorphic Encryption Scheme that can be used for storing data securely in the cloud in order to curb the problem. Online survey was conducted where the researcher used the internet to have access to online materials such as journals; articles and was able to obtain 29 relevant papers for the study. Systematic Mapping Studies approach was used to scrutinize and categorize articles into groups from a larger dataset. The researcher has tested, compared and evaluated the time complexity, execution time and security of the homomorphic encryption schemes, Elgamal, RSA, SAM, SA, SDC and DGHV. As a result, in the calculations of the execution time of Elgamal, SA, SDC, DGHV, RSA and SAM schemes, it was observed that Elgamal is fast, but in terms of time complexity and security SAM is the best. SAM scheme shows that it has a very good complexity of time, its execution time is very fast and security can be trusted because its secret key is represented by a prime number and it is protected when ciphertext is retrieved. Therefore, SAM scheme is the best homomorphic encryption scheme for storing data securely in the cloud.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Cloud computing has given more opportunity to the users of the cloud to store and retrieve their data stored in the cloud where ever and any time they need it. Cloud storage has become an admirable technology in cloud computing environment, which enables users to store their data and access it using any electronic device (Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009).

The cloud in recent days gained popularity in many sectors including both private clients and companies. Storage as a service is a cloud business environment in which storage space in the cloud is provided in the service provider storage infrastructure to people or companies. Hence it is prudent for efficient security measures such as secure schemes for data stored in cloud to ensure data integrity and confidentiality.

Security for cloud data is the most needed requirement because of the increasing number of users using the public cloud for storing data. Security is necessary for preserving the integrity, confidentiality and availability of information and other resources in cloud (William, 2006) as cited in (Payal, Shraddha, Shafica, Niyatee, & Rutvij, 2014).

Encrypting data before putting it in the cloud can help enhance data integrity and confidentiality issue. Nonetheless, verification of data integrity is a tedious job without getting a copy of data or fetching it from the server. Because of this, the cryptographic primitives cannot be used right in place for protecting outsourced data (Syam & Subramanian, 2011).

Besides, an unexperienced way to check the integrity of information kept in the cloud is to download the stored data so that its integrity can be validated, which is unreasonable for excessive I/O cost, high communication over the network and inadequate computing ability. Hence, effective and real mechanisms are desired to defend the confidentiality and integrity of clients' data with lowest computation, communication and storage problems (Syam & Subramanian, 2011).

Although the prevailing schemes target to provide integrity verification for various data storage systems, the confidentiality of data has not been completely solved.

1.2 Background to the Study

Cloud computing marks a new step towards IT infrastructure dematerialization and gets much attention, both in publications and among users. It can be realized that, many people make good use of cloud computing services for their own private desires.

For instance, several individuals use social networking websites or webmail which are all cloud services. Clients of cloud computing are attaining freedom, ergonomics and simplicity.

This new pattern caused the internet a huge depository where resources are networked world wide, simply shared and available to everybody as services. Virtualization is one of the current technologies used to deliver these cloud services.

Virtualization is known to enables you to consolidate your servers and do more with less hardware. It also lets you support more users per piece of hardware, deliver applications, and run applications faster (Armbrust, 2009). These attributes that virtualization hold are the core of cloud computing technologies and is what makes it possible for cloud computing's key

characteristics of multitenancy, massive scalability, rapid elasticity and measured service to exist (Sean & Kevin, 2012).

Virtualization is a set of hardware and software methods that permit to run multiple operating systems at the same time on devices that are completely kept at separate locations from one another. Thus, an operating system called "host" is installed on a machine and hosts operating systems "guests" or "virtual machines" (Yasmina & Rahal, 2015).

Cloud is designed in such a way that it is public to a lot of users and abundant information known as Big Data is kept on a shared platform; therefore, security of data is a high risk. As a result, the need for good security in cloud computing environment to secure huge data stored in it has been emphasized (Kokila & Princess, 2015; Iswarya, 2014).

Cloud security challenges became a huge problem for numerous scholars; the main concern was to pay attention to security, which is the major worry of organizations considering to make a move to cloud. But to adopt the use of the cloud, it implies that the appropriate security concerns are guaranteed. The question many people ask is that, to what extent can we guarantee privacy, secrecy and security of data in cloud computing system? The response to the question is fully homomorphic encryption. Homomorphic encryption permits to make computation over data that was encrypted without having to decrypt it (Yasmina & Rahal, 2015).

Even though the suggested solution has numerous weaknesses, nevertheless has cleared the way for many research work to be carried out on this homomorphic encryption technique. My study is also sailing towards that direction with this work.

1.3 Statement of the Problem

Any cloud user can connect the cloud applications using the internet to access stored data anywhere since cloud provides a way to access it. The data kept in the remote data storage center can be accessed or managed through service providers of the cloud. Stored data in the remote center for processing have to be done with ultimate caution. Safety and security of data in cloud computing is one of the serious concern that has to be looked at currently.

Data is at high risk if security measures are not put in place for data operation and transmission. The reason for this is that, a group of cloud users are provided with cloud computing facilities to access it therefore there is the possibility of having high data risk.

Once data get to the cloud, the users do no more have the ability to control it. This raises some concerns about the challenging issues related to integrity and confidentiality of the data stored in cloud environment. The integrity and confidentiality of the stored data in clouds are of paramount importance for their functionality, therefore rendering the data security as unstable if not properly managed. This concern made the researcher to find out the best advance Homomorphic Encryption Scheme that can be used for storing data securely in the cloud in order to curb the problem.

1.4 Aim and Objectives of the study

The purpose of this study is to critically examine Advance Homomorphic Encryption Schemes used to store data securely in the cloud.

The specific objectives of this study include;

1. To Identify the current advance Homomorphic Encryption Schemes that are used to securely store data in cloud.

2. To investigate which advance homomorphic encryption schemes is the best for storing data securely in cloud.

1.5 Research Questions

The researcher seeks to find answers to the following questions:

1. What are the current advance homomorphic encryption schemes available in the cloud?

2. Which advance homomorphic encryption scheme is the best for storing data securely in the cloud?

1.6 Significance of the Study

This study has both practical and theoretical implications. The findings and outcomes of the study can be used to check the validity of findings of other studies that were conducted previously on similar or the same topic. Again, in the business and marketing field, the study is very important and immensely significant in many ways. This study is also of great significance to the scholars and researchers.

Furthermore, based on these facts, this study explains essential terminologies related to storing and retrieving data in cloud with the target to answer questions frequently asked by users who are not in computer fields.

Additionally, the findings of this project can help both clients and managers of cloud computing to know the best homomorphic encryption schemes to use to store and retrieve data in the cloud securely. Similarly, this research can also influence the administrators, managers and service suppliers of services of the cloud to improve upon the needed services, privacy and security of data in the cloud.

1.7 Limitation

A number of challenges were encountered during the data collection and in the course of writing the dissertation. Some useful information was not made available to the researcher online due to poor network and inappropriate keywords used for searching for information.

The researcher was financially and timely constrained during the collection of data once the researcher had conducted an online survey and made use of both hardware and software resources including network services to download electronic books, articles and journals relevant to the topic.

The inclusion and exclusion criteria adopted for screening of papers to determine which articles to be selected for the purpose of this study, the researcher did not include articles written in other languages apart from English which may contain information relevant to the study. The study is restricted to only homomorphic encryption schemes which may not be a representation of all encryption schemes.

1.8 Delimitation

The study did not take into account all the encryption schemes but only based on Homomorphic Encryption for data privacy and security in cloud.

Due to some constraints such as funds and time, this study did not consider many data security challenges.

The conclusion and generalization would therefore not be applied to all encryption schemes. This implies that for a more thorough study, all encryption schemes for storing data securely in cloud should be considered.

1.9 Definition of Terms

This sub segment contains the portrayal of significant ideas that structure the premise of this study. All the ideas are completely clarified in subtleties with their particular sources later.

Encryption: The way toward encoding messages or data so that lone approved gatherings can understand it.

Cloud computing: Form of cost-proficient and adaptable utilization of IT administrations. The administrations are offered in the nick of time over the web and are paid per utilization.

Clusters: Locally conveyed units with a similar sort of equipment and working frameworks being equipped for preparing a lot of information cooperatively.

Grids: Globally conveyed units with various working frameworks and equipment being fit for preparing a lot of information cooperatively.

Hybrid Cloud: A blend of a private and open cloud.

Instruction as a Service: Users having the option to utilize servers, stockpiling, organize settings on-request from different suppliers on a compensation for every utilization premise.

Platform as a Service: Developers having the option to construct their own applications advertised on improvement stages that are kept up and made sure about by different suppliers.

Private Cloud: Clouds that are utilized in a private system giving greater security.

Public Cloud: Clouds that can freely run anyplace on the planet.

Scalability: Refers to the presentation of dealing with developing measures of work.

Software as a Service: Users can use programming being offered over the web without agonizing over its support, back-ups or security.

Supercomputers: Machines collected with a great deal of processors that are converted into a machine with superior capacities.

Utility Computing: The general thought of figuring assets being offered as an assistance.

Confidentiality: No-one for whom it was unintended can understand the data.

Integrity: Without specifying the alteration, data cannot be turned away or travel between sender and intended beneficiary.

Non-repudiation: The data maker / sender can't deny their expectations when creating or transmitting the data at a later stage.

Confidentiality: Nobody for whom it was unintentional may comprehend the data.

Authentication: The sender and receiver will claim the identity of each other and the data's cause / goal.

1.10 List of Abbreviations

NIST	National Institute of Standards and Technology
RSA	Rivest, Shamir, Adleman
PHE	Partial Homomorphic Encryption
FHE	Fully Homomorphic Encryption
IaaS	Infrastructure-as-a-Service
PaaS	Platform -as-a-Service
SaaS	Software-as-a-Service
SLA	Service-Level Agreements
RIA	Rich Internet Applications
CSA	Cloud Security Alliance
TCP/IP	Transmission Control Protocol/Internet Protocol
SWHES	SomeWhat Homomorphic Encryption Scheme

BES	Bootstrappable Encryption Scheme
MPC	Multi-Party Computation
HAE	Homomorphic Authenticated Encryption
SMS	Systematic Mapping Studies
SLR	Systematic Literature Review

1.11 Organization of the Study

This research is made up of five (5) five chapters. Chapter one comprises introduction that includes background to the study, Conceptual Frame Work, Research Process, Statement of the problem, Purpose of the study, Objectives of the study, Research Questions, Significance of the Study, Delimitation of the Study, Definition of Terms and Organization of the Study.

The second chapter reviewed Literature on works related to my study done by other researchers under sub-headings, Cloud Computing, Homomorphic Encryption.

The third chapter is concerned of the Methodology which comprises the Introduction, Research method, Justification of the Approach chosen, Search Strategy, Data Analysis and Classification, Threats to Validity and conclusion.

The fourth chapter consist of the discussion of findings.

The fifth chapter covers Introduction, Summary, Conclusion, Recommendation and suggestion for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter emphasizes on what earlier researchers and authors have written and said about this topic. Here, I consulted published articles, journals and web sites for materials relating to my study. This chapter is very key for my dissertation because the preliminary search was helping me to convey and advance my research ideas. Cloud computing help clients to enjoy some potential benefits such as scalability, instant availability and sharing of resources because cloud computing provided them with boundless computing power. Adoption of cloud computing services by users is slow due to issues of loss of data privacy even though cloud has become a mature service model.

Data encryption is the solution to this issue, but then if the users wish to work on their data that is encrypted in the cloud, he/she have to disclose his/her secret key to the service provider in order to decipher the encrypted information before executing the necessary operations (Lauter, Naehrig, & Vaikuntanathan, 2011).

To solve cloud computing security problems, Homomorphic Encryption is the proper remedy to problems concerning security of cloud computing, once the scheme aids to execute the manipulation of the ciphertext having to give out the decipher the data.

Baohua and Na (2014), recommended an enhancement of the second scheme of Gentry to make cloud more secure by applying fully homomorphic encryption in cloud.

This study addresses cloud data storage issues and problems related to confidentiality and security of user' data. The core goal of this study is to present the concept of encryption that is fully homomorphic and the means to make use of these concepts to provide security to data stored in cloud.

2.2 Conceptual Framework

The conceptual framework of this study shows the five-layer cloud architecture to accomplish the objective of storing data securely in the cloud.

According to Arokia, Paul, and Shanmugapriyaa (2012), Cloud data storage architecture is basically about delivery of storage on request in an extremely climbable and multi-tenant way. Generally, cloud storage architecture comprises a front end that carry an API across to have access to the storage.

In the old storing system, the SCSI protocol is the API; however, these protocols are evolving in the cloud. The file-based front ends, web service front end all together with the traditional front end are all found in the cloud. The storage logic also known as the middleware layer is found behind the front end (Spoorthy, Mamatha, & Santhosh, 2014).

A lot features such as data duplication and data reduction, over the traditional data placement algorithm are implemented in this layer. The back end is responsible to implement the physical storage of data. This could be an internal protocol implementing specific features or a traditional back end to the physical disks.

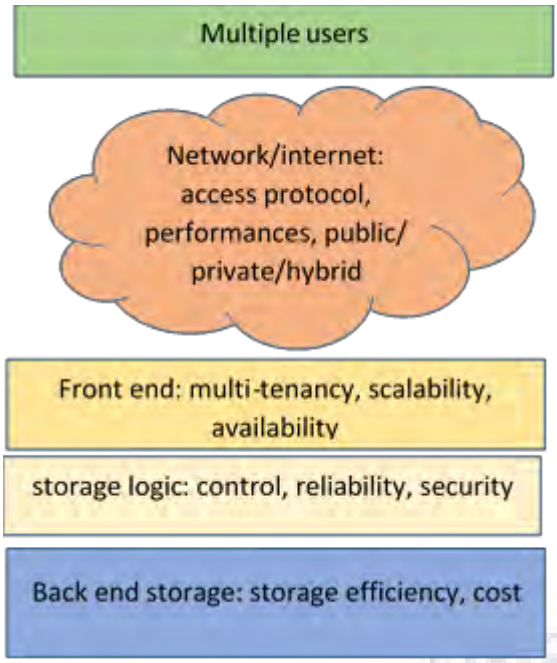


Figure 1: structural design of cloud storage (Spoorthy et al., 2014)

The figure 1 above shows a sample of the physical features displaying the recent cloud storage structural design.

2.3 Encryption

In cryptography, encryption is the way toward changing over a message or messages or information/data into the structure that lone approved clients can peruse and get it. Encryption is the way toward changing data/information into the structure called ciphertext which unapproved individuals cannot easily understand (Nitin et al., 2012).

According to Kerckhoffs (1883), security does not only rely on how complex a code is but rather rely on the secrecy of the scheme. Encryption only prevents the message content to be understood by the interceptor but does not prevent interception. An encryption algorithm is used to encrypt a message (data) or information known as plaintext to get a ciphertext that can be read and make meaning to the user if decrypted. It is for a significant and specialized reasons that

each encryption plot typically utilizes a pseudo-irregular encryption key produced by a calculation. Despite the fact that it tends to be conceivable to decode the scrambled message without having the key however for encryption plots that are very much evolved or structured, a colossal measure of computational abilities and assets are required. Cryptography in this ongoing year frets about the accompanying four goals (Ayub, Manish, & Monjul, 2015).

Confidentiality: The message must be kept secret and cannot easily be understood by an unauthorized person to whom was not intended.

Integrity: Integrity: the message must stay as it was and can't be changed away or when send between the sender and the proposed recipient.

Non-repudiation: The sender or the generator of the message or the beneficiary of the message can't later deny their goals of sending or getting a given message.

Verification: The generator of the message or the beneficiary of the message can approve, confirmation and affirm their characters and the source and the goal of the message.

A cryptography scheme is thought to be publicly known while the mystery snippet of data, for example, key is liable for the mystery of the scheme. There are two kinds of encryption schemes as indicated by key management. The two kinds are: Symmetric and Asymmetric encryption schemes (Nitin, Saibal, Pal, Dhananjay, & Upadhyay, 2012).

2.3.1 Symmetric Encryption

An encryption framework where the sender and recipient of a message share a solitary basic key that is utilized to encode and decode the message is called Symmetric Encryption. Symmetric-key frameworks are quicker, yet their fundamental disadvantage is that two parties wishing to impart need to trade the key in a protected manner. Furthermore, scalability is issue as the

quantity of clients' increment in the system. Because of its mystery nature, symmetric-key cryptography is some of the time alluded as mystery key cryptography (Nitin et al., 2012).

2.3.2 Asymmetric Encryption

An encryption system is called asymmetric encryption on the off chance that it utilizes two keys rather than one key as in symmetric encryption. One key encodes the information and the different decodes. It is additionally variably alluded to as open key cryptography. A significant component of the open key structure is that general society and secret keys are connected so that solitary the open key can be used to encrypt data and its comparing secret key can be employed to unscramble the message (Nitin et al., 2012).

Also, it is for all intents and purposes difficult to find the secret key regardless of whether the open key is known. Open key cryptography was designed in 1976 by Whitfield Diffie and Martin Hellman (Menezes, Van Orschot, & Vanstone,1997; Van Tilborg, 2005), and the plan was called Diffie-Hellman encryption. Security of this kind of plan depends on difficult issues in arithmetic, which are hard to illuminate in polynomial time. In any case, the drawback is that they are slower than the symmetric plans due to non-insignificant numerical calculations. That is the reason this encryption conspire is utilized uniquely for encryption of little information or keys while symmetric plan can be utilized for bigger ones.

2.4 Cloud Computing models

As indicated by the Institute of Standard and Technology (NIST), distributed computing is a design to enable suitable network services and good accessibility to a mass computing programs

and resources that can be frequently provided and released with negligible managing effort or service provider contact.

This cloud model was made up of five (5) important features, three service structural designs and four deployment models (Mell & Grance, 2011).

Cloud services permit people, industries and business organizations to make use of software and software to be managed by cloud service providers.

Cloud computing was built around the idea generated on a basic principle of re-usability of information technology abilities.

2.4.1 Cloud computing service models

2.4.1.1 Infrastructure-as-a-service (IaaS)

This administration model speaks to the joining between the cloud facilitating stages and assets. Couple of them convey both the oversight/the board layer/layer and the physical foundation; others convey just the administration. In different cases, the administration layer is fuse with IaaS arrangements that convey physical framework and increment the incentive for them. IaaS arrangements are fitting to delineate the framework foundation, yet have constrained administrations for creating applications. IaaS is offering to the client a cloud bundle that is giving command over an IT framework. Services and facilities such as servers, storage and network settings are cared for by the cloud service providers, despite the fact that customers have virtual instance of that (Armbrust et al., 2009; Buyya et al., 2009).

2.4.1.2 Platform-as-a-Service (PaaS)

PaaS has the framework, being a part of the administration offered to the customer. The Pure PaaS is offered distinctly to the customer - level middleware and at long last it has to have a virtual or physical structure. PaaS gives the office to help the advancement lifecycle from structure, usage, troubleshooting, testing, arrangement, activity and backing of rich web applications (RIA) and online administrations. Here for the most part the web program will be utilized for the turn of events (Armbrust et al., 2009; Buyya et al., 2009). With PaaS a whole programming condition can work/execute at a cloud specialist co-op while not agonizing over the innovation underneath it. The database and the application environment needs to be taken care of (Armbrust et al., 2009).

2.4.1.3 Software-as-a- Service (SaaS)

The majority of the applications here are online applications that relying upon the cloud for offering types of assistance for the end client. The autonomous programming vendors can make use the web to convey their computer generated software foundation as an assistance and stage as a help. Organizations can utilize programming that is made accessible online on a rental or utilization premise as opposed to purchasing the entire programming bundle locally without being certain whether the venture will pay off on a drawn-out premise. The product supplier will deal with all support or update gives that may happen (Armbrust et al., 2009).

Everything-as-a-Service (XaaS) is one of the most indispensable part of distributed computing condition, since administration of the cloud offered by different specialist organizations can consolidated between them to offer an "response covering the entire processing heap of the structure" (Buyya et al., 2013).

2.4.2 Cloud computing deployment models

2.4.2.1 Private Cloud

This model is generally set up inside an association' datacenter. Private cloud is another term that a few merchants have as of late used to portray contributions that copy distributed computing on private systems. Assets that are adaptable and computer-generated programs that the cloud sellers gave are joint together and can be accessed by cloud users and use in the private cloud. Just the association and picked partners may approach work on a particular Private cloud (Kamal, Kattit, & Elmarraki, 2014).

2.4.2.2 Public cloud

A public/open cloud is a model which offers customers the chance to approach the assets, administrations and framework and are given off-website over the Internet (Kuyoro, Ibikunle, & Awodele, 2011). It's chiefly relies upon a compensation for each utilization model, much the same as that of a prepaid power meter perusing framework consistently adaptable enough to deal with spikes in demand for cloud enhancement. Open mists are handled or overseen by clients or client over the Internet. Open mists are not completely made sure about when contrasted with other cloud models with the explanation that it represents another weight of ensuring that there will be no risky assaults on all applications and information got to on the open cloud. In any case, all security and control issues must be all around arranged and be set up to cultivate security controls out in the open cloud (El Marraki, Hedabou, Belkasmi, & Kartit, 2016).

2.4.2.3 Hybrid Cloud

Hybrid cloud consolidates both open and private cloud models that endeavors to unravel the impediment issues of each approach. This is another idea that joins assets from both inward and outer suppliers will be the most will turn into the most mainstream decision for undertakings. The hybrid cloud work so that a segment of the foundation runs openly mists whiles the other part runs in the private mists. As far as adaptability is concerned, hybrid is more adaptable than both private and open cloud. To be explicit, half breed cloud guarantees an exceptionally close control and security over application information when contrasted with open cloud, while as yet encouraging on-request administration development and compression. On the drawback, planning a half and half cloud requires cautiously deciding the best split between open and private cloud parts (Chunye, Qiang, Haitao, & Zhenghu, 2010).

2.4.2.4 Community cloud

The people group permits framework to be shared by numerous or a few foundations or associations and might be handle or managed within or by a service provider. This model is only here and there advertised. It unites, by and large, the structures with similar intrigue (generally security) and might be in a similar arena of action (SO & kuyoro, 2011).

2.5 Security of Cloud Computing

Cloud computing security comprises concepts which includes network security, all facilities, equipment, and appropriate control strategies put in place to protect data, applications and infrastructures of the cloud. the cloud is perceived to be made up of interconnection of several things which made it complex and problematic and hence its environment needed to be

safeguarded. Issues of security in cloud computing platforms can cause financial damage and an evil status if the cloud environment is concerned with large public. Any harm caused to users' information through an illegal intruder is not accepted (El Marraki et al., 2016). Two ways by which information in the cloud can be attacked are insider attack and outsider attack. The insider is someone who knows much about the system and usually within the system. For example, the administrator knows the system well and can possibly hack the customer data. It is very cumbersome to identify insider, therefore, customers need to be very cautious when storing data in cloud. Due to this, there is the need to call for methods/measures that will protect the data. Although, a service provider/third party can gain entry into users' data, he should not fetch the actual data. So, it is good for users' data to be encrypted before sending to the cloud storage (Arockiam & Monikandan, 2013).

Technologies and their standardization that are developed makes some algorithms and protocols available for answering these issues. From NIST view, some concern barriers such as security, interoperability and portability prevents the fast acceptance of cloud. In order to preserve total security for data in cloud, is necessary to meet the safety requirements (confidentiality, integrity, accessibility and non-repudiation) at each level. In data life cycle, it is very important for security to be involve at every level (Yasmina & Rahal, 2015).

For issues relating to security in cloud for example, third party control, privacy and data security to be handled efficiently, all data in cloud needed to be encrypted using the traditional cryptosystems (Aderemi, Atayero, & Oluwasey, 2011).

Cloud user need to give their undisclosed/private key to the cloud provider before he can execute any necessary computation on the coded information. The service provider need to first decipher the information to perform required actions and send the outcome to the customer. To find a

remedy to this security issues, it very crucial to employ the use of Homomorphic Encryption to encrypt users' information.

2.5.1 Cloud Computing Security Framework

Lately, it is essential to construct cloud security frame work because recently there are several security difficulties related to cloud computing that are blocking the growth and commercialization to cloud computing. Figure 2 below is a cloud computing security structure.

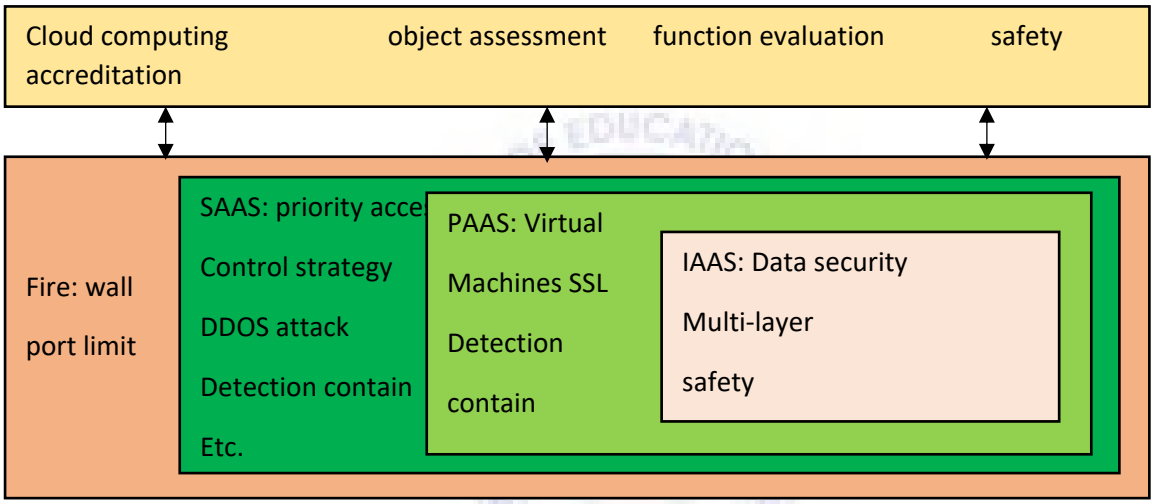


Figure 2: Cloud Computing Security Framework (Xiaowei, Xiaosong, Ting, Hongtian, & Xiaoshan, 2012)

2.5.1.1 Firewall

Firewall is essential for distributed computing. It really upgrades the protection enormously The strategy is to reduce the type of open port. Among every one of them, the Web server bunch opens port 80 (HTTP port) and 443 (HTTPS port) to the world, application server bunch just open port 8000 (uncommon application administration ports) for the Webserver group, database server group just open only port 3306 (MySQL port) for application server gathering.

Simultaneously, the three system server groups open port 22 (SSH port) for clients, and other system association decline by default. The safety will be enormously improved by this part (Bikram, 2009).

2.5.1.2 Security Measures of SaaS

SaaS suppliers give customers complete application and parts in distributed computing, and should ensure security for the system and segments. The capacities for defense have two main perspectives:

Need to get to control procedure: SaaS suppliers offer confirmation of personality and access control work, usually the name of the client and the component of secret word check. Clients should have sufficient knowledge of the supplier they have picked to remove the danger to the security of the interior elements of cloud applications. At the same time cloud suppliers should give high strength, change the password on schedule, make password length based on sensitive degree information and should not use the function, for example, old secret word to strengthen the security of the user account (Xiaowei et al., 2012).

Normal system assault anticipation: relying on the absence of experienced system assault protection measures, for DDOS assault, given its assault implies, suppliers can use a few strategies: for example, designing a firewall, obstructing the ICMP and any obscure convention; closing superfluous TCP / IP administrations, arranging firewalling to deny Internet requests. Suppliers can routinely screen TCP administration for use-type assault, and update programming patches in time. The conventional system assault has been read for quite some time and it is possible to use extremely developed items, cloud suppliers can use these items to guarantee the security of the cloud (Boss, Malladi, & Quan, 2007).

2.5.1.3 Security Measures of PaaS Layer

In distributed computing, PaaS is the center layer, the safety efforts are two viewpoints. Virtual machine innovation application: Using the upsides of virtual machine innovation, suppliers can set up virtual machine in existing working framework. Simultaneously, set access limitations, basic clients can work PC equipment just through advancing working authorizations. This is acceptable recognized the customary clients and executives, regardless of whether the client has been assaulted, there will be no harm to the server (Xiaowei et al., 2012).

SSL assault shielding: For the conceivable presence of SSL assault, the client must reinforce forestall technique. Suppliers ought to give the relating patch and measures, so the client can fix in the first run through, and ensure the SSL fix can rapidly work. Simultaneously, utilizing the firewall to close some port to forestall regular HTTPS assaults, reinforcing the board authority, making security testament difficult to get are acceptable guarding techniques (Jamil & Zaki, 2011).

2.5.1.4 Security Measures of IaaS Layer

By and large, the information encryption is not simply reliable for the mix of different client information, but suppliers also need to isolate client information placed on different information servers to reduce the productivity of information (Zhang & Chen, 2010). Separating the user data storage can prevent data separation chaos. For information reinforcement, significant and classified information ought to be supported up, simultaneously, regardless of whether there is sure equipment disappointment, information can be handily recuperated and the recuperation time likewise needs an assurance.

2.6 Homomorphic Encryption for Data security and privacy in Cloud

The difficult issue emerges if there is a requirement for registering freely with private information or to alter capacities or calculations in a way that they are as yet executable and their security is ensured. Homomorphic encryption is perfect to be utilized once it permits calculation to be done on encoded information. Among distributed computing attributes, the sharing of protection structures and information handling, one issue of this is the conservation of classification among customer and supplier. To solve the problem encryption could be utilized, since the client can decide to store just encoded information (Yasmina & Rahal, 2015).

The issue is that while information can be sent to and from a cloud supplier's server farm in scrambled structure, the servers that power a cloud can't accomplish any work on it that way. So if the customer needs to perform computations on its information in the cloud, the mystery key to decode the information ought to be imparted to the supplier. Sharing the key would permit the cloud supplier access to the information. The response to this issue is the homomorphic encryption. The customer would furnish the cloud with executable code to permit it to chip away at the information without unscrambling it (Yasmina & Rahal, 2015).

The outcome will be returned to the customer with everything scrambled. So since the customer is the main holder of the mystery key, nobody else can decode neither information nor results.

An encryption plot has three parts (KeyGen, Enc, Dec):

• KeyGen: the capacity that creates pair of keys (open key pk and mystery key sk).

• Enc: an encryption calculation that takes the open key and the plain content to tomb M and gives the ciphertext.

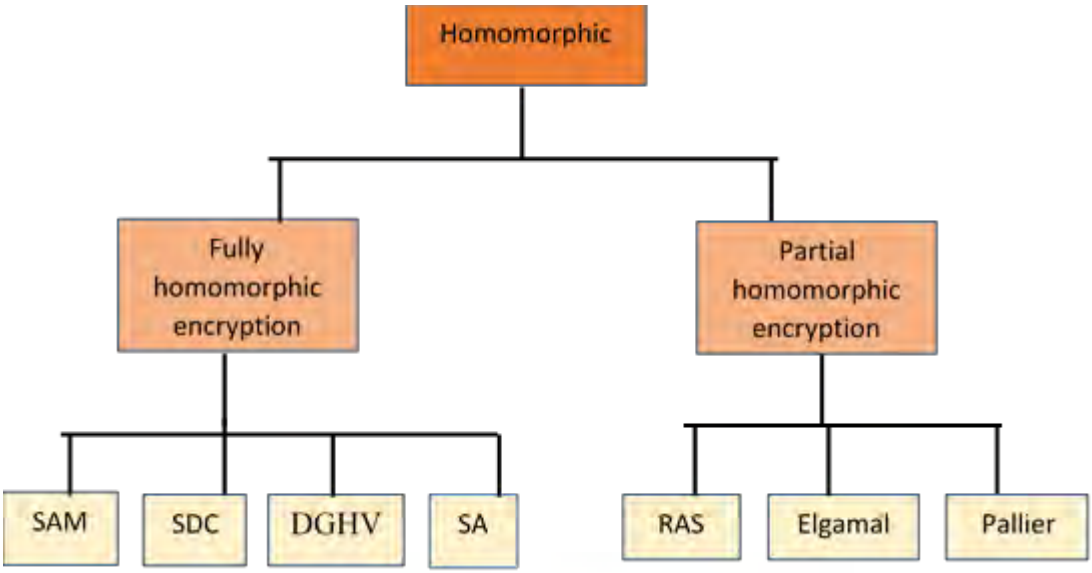
• Dec: an unscrambling calculation that takes the ciphertext c and the mystery key and recoups the plain content M (Yasmina & Rahal, 2015).

2.6.1 Categories of Homomorphic Encryption

Homomorphic encryption scheme has been arranged into two main groups. Those that have either additive or multiplicative property (partially homomorphic encryption). They permit either addition or multiplication to be performed on scrambled information. Instances of partially encryption plans (PHE) incorporate Paillier, RSA, Elgamal and some more (Ihsan & Saad, 2016).

Those that have both addition or multiplication property (completely homomorphic encryption). They permit both expansion and augmentation to be performed on scrambled information. Instances of completely encryption plans (FHE) incorporate SAM, SDC, DGHV (Shihab & Ali, 2018).

Structuring an encryption plan to help both the expansion and duplication plot simultaneously was slippery. Boneh, Goh, and Nissim (2005) made a decent attempt and moved nearer, permitting a few expansion activities and just a solitary increase. In 2009, Gentry buckled down and had the option to tackle the issue in comparable work. Upper class' work was fruitful since his plan can bolster both expansion and increase at the same time in completely homomorphic encryption



Source: Researcher's Field Work, 2020

Figure 3: Categories of Homomorphic Encryption

2.6.1.1 Partially Homomorphic Encryption

An encryption scheme can be considered to be partially homomorphic if it shows either additive or multiplicative properties and not both. It is obvious that PHE is very useful in some certain applications. In addition, for some practical applications, the efficiency of some PHE is very high (Ihsan & Saad, 2016).

2.6.1.1.1 The RSA Encryption Scheme

RSA is an early case of PHE and presented by Rivest, Shamir, and Adleman (Rivest, Shamir, & Adleman, 1978) soon after the creation of open key cryptography by Diffie and Helman (Diffie & Hellman, 1976). RSA is the principal possible accomplishment of the open key cryptosystem. In addition, the homomorphic property of RSA was presented by Rivest, Shamir, and Adleman (Rivest et al., 1978) soon after the fundamental work of RSA. For sure, the primary confirmed

utilization of the expression "protection homomorphism" is presented by Rivest, Shamir, and Adleman (Rivest et al., 1978). The security of the RSA cryptosystem depends on the hardness of figuring issue of the result of two huge prime numbers. The delineation beneath shows a calculation of RSA performing multiplication (Rivest et al., 1978).

Key generation

Choose two huge primes primes p and q, such that $p \neq q$.

n=p*q.

φ(n)=(p-1) *(q-1), where φ is Euler's totient function.

Select an integer e such that $1< e < \varphi(n)$ and gcd (e, φ(n)) = 1 (coprime).

d= e-1mod φ (n)

public key-------(e, n)

private key------- d

Encryption

C = me mod n

Decryption

m= cd mod n

The multiplicative homomorphic property of RSA scheme

is as follows (Eyad, 2015):

Given C1= m1e mod n, C2 = m 2 e mod n

c1.c2= Epk(m1). Epk(m2)

= m1e. m2e (mod n) = (m1. m2) e (mod n)

= Epk(m1.m2)

B. Additive Homomorphic Schemes

A Homomorphic Encryption is additive, by being able to compute Enc (x + y) from Enc(x) and Enc (y) with an algorithm deprived of knowing x and y (Xing, Chen, Zhu, & Ma, 2006), such as Paillier and Goldwasser-Micali algorithms.

2.6.1.1.1.1 Time Complexity of RSA Cryptosystem

Let n is the size of input message that in the type of decimal digits.

Ciphering function:

C=Me mod (n)

This implies:

T (C) = O ((log (n))3) bit operation.

T (Encryption) = O ((log (n))3) bit operation, according the most costly operation.

Also,

T (Encryption) = O (log (n)) arithmetic operation.

Deciphering function:

M= Cd mod (n)

This implies:

T (M) = O ((log (n))3) bit operation.

T (Decryption) = O ((log (n))3) bit operation, according the most costly operation.

Also,

T (Decryption) = O (log (n)) arithmetic operation.

2.6.1.1.2 Elgamal Encryption Scheme

New public key encryption scheme which is the improved version of the original Diffie-Hellman Key Exchange (Diffie & Hellman, 1976) algorithm suggested by Taher Elgamal in 1985, which is based on the hardness of certain problems in discrete logarithm. It is mostly used in hybrid encryption systems to encrypt the secret key of a symmetric encryption system.

Let (G, q, g, h) be the public key in G group, where h=gx and x is the secret key, the encryption of a message m is

Ɛ(m) = (gr, m.hr) for r ϵ {0,1,…q-1}

The homomorphic property is then

Ɛ(x1). Ɛ(x2) = (gr1, x1.hrl) (gr2, x2.hr2)

= (gr1+r2, (x1. x 2) hr1+r2)

= Ɛ(x1.x2)

2.6.1.1.2.1The time complexity of Elgamal Cryptosystem

If n is the size of the message input in decimal digits.

Ciphering function:

K= Yr(mod p)

C1 = gr (mod p)

C2 =M. K (mod p)

Then,

T (K) = O ((log (n))3), (Modular exponentiation)

T (C1) = O ((log (n))3)

T (C2) = O (2(log (n))2) = O ((log (n))2)

Where, (log2 n) is the number of bits of n.

T (Encryption) = O ((log (n))3) bit operation, according the most costly operation. Also,

T (Encryption) = O (log (n)) arithmetic operation.

Deciphering function:

K= C1x (mod p)

M = K-1. C2 (mod p)

Then:

T (K) = O ((log (n))3)

T (M) = O (2(log (n))2) +T (K-1)

T (K-1) = O ((log(n))3), (by extend Euclid's method)

T (M) = O (2(log (n))2) + O ((log(n))3)

T (M) = O ((log (n))3) bit operation

T (Decryption) = O ((log (n))3) bit operation, according the most costly operation.

Also,

T (Decryption) = O (log (n)) arithmetic operation.

2.6.1.2 Fully Homomorphic Encryption

A cryptosystem is known as completely homomorphic if it has both the additive and the multiplicative properties of homomorphism. The first (and only currently) above listed framework is a lattice-based cryptosystem that was introduced and developed by Craig Gentry in 2010. In the words of Gentry (2009), FHE is considered much more effective and a great way to efficiently secure the outsourced data. The scheme proposed by Gentry contains three important components:

(1) A very homomorphous encryption scheme(SWHES)

(2) Bootstrappable (BES) encryption scheme

(3) A combination of two elements above

This scheme possessed the capacity to perform homomorphic computation on low degree polynomials. Completely homomorphic encryption is even more efficient. Such a scheme allows to build programs for any desired features, which can be performed on encrypted inputs to produce a result encryption. Since a program like this never needs to decrypt its inputs, an untrusted party can run it without disclosing its inputs and internal state. For example, in the context of cloud computing, the existence of an efficient and fully homomorphic cryptosystem would have great practical implications in outsourcing private computations.

At what time Craig Gentry suggested the first possible creation of a completely homomorphic scheme in 2009 (Gentry, 2009). In Boolean algebra, Gentry 's work conducting addition and multiplication at the concurrently relates to “AND (^)” and “XOR (⊕)”. The incredible mark of performing these two Boolean functions is that any calculation can be converted into a function comprising of only (^) and (⊕). To convert a function into an easy one in algebra, several

techniques can be used. When you use this technique, you can transform a function to use a particular Boolean operation only. (e.g. ∧ or ⊕). For example, ¬A can be expressed as A ⊕ 1, another example is A ∨ B, this can be converted into (¬A) ∧ (¬B), then converted into (A ⊕ 1) ^ (B ⊕ 1). By utilizing such techniques, all functions can be converted into a series of (∧) and (⊕) operations. This is the basis of Gentry’s work (Eyad, 2015).

Gentry uses cryptography based on lattice. Its projected completely homomorphic encryption comprises numerous stages: starting with what was referred to as a somewhat homomorphic

encryption scheme by means of ideal lattices, restricted to the evaluation of low-grade polynomials over coded information. It is restricted because, to some extent, each coded text is noisy, and the noise increase as one adds and multiplies encrypted texts, till at the end of the day the resultant encrypted text is unclear. Next, it squashes the decryption process so that it can be expressed as a low-level polynomial supported by the scheme. Finally, a bootstrapping conversion is useful, through a recursive self-embedded procedure, to get a completely homomorphic scheme (Jaydip, 2013).

2.6.1.2.1 DGHV Encryption Scheme

Van Dijk, Gentry, Halevi, and Vaikuntanathan (2010) implemented an absolutely homomorphic encryption (the so-called DGHV). Their scheme (somehow homomorphic) does not use ideal lattices over polynomial ring, but uses addition and multiplication over integer. This pattern is conceptually easier as compared to the ideal Gentry lattice-based scheme, but in terms of homomorphic operation and performance they look similar. In cloud computing DGHV scheme is insecure in the sense that it requires passing the secret key p to the server since the R cipher retrieval algorithm (Li, Song, Chen, & Lu, 2012). DGHV Scheme Description is as follows:

KeyGen (λ): create the secret key p is an odd η-bit integer, $p \epsilon [2\ \eta\text{-}1, 2\ \eta)$.

Encode (pk, $m\epsilon\{0, 1\}$): to encode a 1-bit message m: creates a big multiple of the private key, e.g., p q, a minor even number 2r where r is the noise ($r < p/4$ or $2r < p/2$), $r \approx 2$ and create $q \approx (2\ \eta)\ 3 \rightarrow c = p\ q + 2r + m$

Evaluate (pk, C, c1, . . . , ct): assumed that the (binary) circuit C with t inputs, and t ciphertexts ci, apply the (integer) addition and multiplication gates of C to the ciphertexts, executing all the operations over the integers, and return the resulting integer

Decrypt (sk, c): to decrypt ciphertext c

$m = (c \bmod p) \bmod 2$

Proof:

Let $c1 = pq1 + 2r1 + m1$, $c2 = pq2 + 2r2 + m2$

Additive Homomorphism:

$c1 + c2 = (q1 + q2)\, p + 2(r1 + r2) + (m1 + m2)$

$[(c1 + c2) \bmod p] \bmod 2 \equiv m1 \text{ XOR } m2 \pmod 2$

Multiplicative Homomorphism:

$c1.c2 = (q1q2p + 2q1r2 + q1m2 + 2q2r1 + q2m1)\, p + 2(2r1r2 + r1m2 + r2m1) + m1m2$

$[(c1.c2 \bmod p) \bmod 2] \equiv m1 \text{ AND } m2 \pmod 2$.

2.6.1.2.1.1Time Complexity of DGHV scheme

Let n be the size of input message unit.

Encoding function:

$c = m + 2r + p^* q$

Then: $T(c) = O(n) + T(2r) + O(n2)$

$T(2r) = O(n)$, by shift operation

$T(c) = O(2n) + O(n2) \equiv O(n2)$ bit operation.

Decoding function:

$m = (c \bmod p) \bmod 2$

Then: $T(m) = O(n2)$ bit operation.

2.6.1.2.2 SDC Encryption Scheme

In the year 2012, Jian Li, Danjie Song et al. proposed a simple FHE based on the Gentry cryptosystem to guarantee secrecy in cloud data storage. Neither DGHV nor Gen10 schemes have shown encrypted text recovery algorithms. The scheme is called SDC (Li et al., 2012). The explanation of SDC scheme as follows:

KeyGen(p): The key p is a random odd integer of P-bit.

Encode (p, m): To encode a bit m ϵ {0, 1}

c = m + p + r * p * q

Where r is a random number of R-bit and q is a constant Q-bit big integer.

Decipher (p, c): Output (c mod p).

Recovery(c):

R = (ci – cindex) mod q.

Any time the client wants to recover contents mindex, he encodes the Keywords

cindex = mindex + p + r * p * q

And conveys cindex to the server.

On getting cindex, server reads the decrypted texts, calculating

R = (ci - cindex) mod q,

once R =0, decrypted text recovery succeeds, and ci is the preferred outcome.

In the SDC scheme, simply transferring q to the server, the server can effectively finish ciphertext recovery procedure without leaking plain text, because the decoding procedure uses the secret key p while the recovery procedure uses the integer q, which is totally dissimilar. This solves both the need for coded text recovery and security of data (Li et al., 2012).

Jian Li, Danjie Song et al. proposed the following proof of the exactness of the structure: Given two messages m1 and m2. The coded text of these messages after encoding:

c1 = m1 + p + r1 * p * q.

c2 = m2 + p + r2 * p * q

To check additively homomorphic property:

c3 = c1+ c2 = (m1 + m2) + (r1 + r2) * p * q * 2p.

m3 = c3 mod p = m1 + m2.

Then SDC scheme has additively homomorphic property.

To check multiplicatively homomorphic property:

C4 = c1* c2 = m1 * m2 + (m1 + m2+ p) p + r1 (p + m2 + r2)p q + r2 (p + m1)p q.

M4 = c4 mod p = m1 * m2.

Then SDC scheme has multiplicatively homomorphic property.

2.6.1.2.2.1Time Complexity of SDC scheme

Let n be the size of input message unit.

Encoding function:

c = m+ p+ r *p*q

Then: T (c) = O (n) + O (n) + O (2(n2))

T (c) = O (2(n)) + O (2(n2)) ≡ O (n2) bit operation.

Decoding function:

m = c mod p

Then: T (m) = O (n2) bit operation.

2.6.1.2.3 SAM Encryption Scheme

Totally homomorphic Encryption by Prime Modular Operation (SAM Scheme). The SAM scheme allows a character to be entered straight from the text into the ciphering equation without transforming a character in plain text to a binary form before entering the ciphering equation (Shihab & Ali, 2018). The SAM pattern accepts the character straight (ASCII character code) into the ciphering

equation: c= m+ r p+ p. q, where c is the coded text, m ∈ [0, p-1], p is the prime big integer, r is the noise and q is the constant big integer, resulting in one coded text for each character in the plain text.

Key Generation

Create a prime big integer p

Create q ∈ Zn

Create r ∈ Zn

Encoding

c =m+ r . p + p . q, where m ∈[0, p- 1]

Decoding

m =c mod p

Evaluation

Assume there are two encrypted texts:

c1 = m1+ r1 p + p.q

c2 = m2+ r2 p + p.q

c3 = c1+c2 → m3 = Dec (c3) (or m3= m1+ m2)

c4= c1. c2 → m4 = Dec (c4) (or m4= m1. m2)

Let c= f (c1... ci), such as:

c= [(c1. c3) + c2] .c4

Let m= f (m1... mi), such as:

m= [(m1. m3) + m2] .m4,

Where f is any function (Addition and Multiplication) applied on the encrypted texts or messages.

c mod p ≡ m, where m <p, otherwise that we must take (m mod p)

The explanation of SAM scheme is as follows:

KeyGen (λ): create the private key p to be a prime big integer, which is selected randomly.

Encode (pk, m ϵ[0, p- 1]): to encode a message m

The encrypted text c = m + r p +p.q, where r is the noise and r is a random big integer and q is a constant big integer.

Evaluate (pk, m1 . . . mt, c1, . . . , ct): use addition and multiplication to t encrypted texts ci, and then decipher the outcome of ci, we obtain an integer number which is similar as the integer number that is a result of applying addition and multiplication to t input mi Decrypt (sk, c): to decode ciphertext c: m = c mod p.

To verify that the SAM pattern approves additive and multiplicative homomorphism:

Suppose, c1 = m1+ r1 p + p.q, c2 = m2+ r2 p + p.q

Additive Homomorphism:

c3 = c1 + c2 = (m1 + m2) + (r1 + r2) p + 2 p.q

m3 = (c1 + c2) mod p = m1 + m2

Multiplicative Homomorphism:

c4 = c1.c2 = m1.m2 + (m1+ m2+ p.q) p.q + r1 (m2+r2 + p.q)

p + r2 (m1+ p.q) p. → m4 = (c1. c2) mod p = m1. m2.

SAM Example:

Select a prime number p =1207645633, and q = 1155942797

Next choose two random integers r1=1828989585 and r2 =1136953862, and

two messages m1 = 65 and m2 = 66

Now calculate c1:

c1 = m1+ r1 p + p.q

c1=65+1828989585*1207645633+1207645633*1155942797

c1=3604740555922587871, (65 is ASCII code of character A).

Now compute c2:

c2 = m2+ r2 p + p.q

c2 = 66 + 1136953862* 1207645633 + 1207645633 *

1155942797, (66 is ASCII code of the character B)

c2 =2769006637161640213

Additive Homomorphism:

Let the addition of two encrypted messages be c3:

c3 =c1+ c2

=3604740555922587871+2769006637161640213

= 6373747193084228084

Now decipher c3:

m3= c3 mod p→ m3= 6373747193084228084 mod 1207645633

m3=131, this is equal to m1 + m2 (i.e. 65 + 66 =131)

Multiplicative Homomorphism:

Let the multiplication of two encoded messages be c4:

c4=c1.c2 = 3604740555922587871 * 2769006637161640213

= 9.98155052459538650428376534197 97e+36

Now decipher c4:

m4= c4 mod p→ m4= c4 mod 1207645633

m4= 4290, this is equal to m1 * m2 (i.e. 65 * 66 = 4290)

2.6.1.2.3.1Time Complexity of SAM scheme

Let *n* be the size of input message, *n* – decimal digit

Encryption function:

c = m + r* p + p*q

Then: T (c) = O (2(log (n))) + O (2(log (n)) 2)

T (c) ≡O (log (n) 2).

Decryption function:

m = c mod p

Then: T (m) =O ((log (n)) 2),

Where, (log2 n) is the number of bits of n.

2.6.1.2.4 SA Encryption Scheme

The SA encryption scheme is mainly based on Euler's theorem. SA is a symmetric (or secret key) encryption scheme and supports both multiplicative and additive homomorphism property. That means it is Fully Homomorphic Encryption Scheme (Shihab & Ali, 2018). The description of SA scheme as follows:

Key Gen: choose two prime numbers p and q

Calculate: S = p *q

Encryption: the message M always will be less than < n, choose random big integer r.

C = M r. p – r +1 mod S, where C is a ciphertext.

Evaluate: apply multiplication and addition on coded texts Ci, and then decode the end result of Ci, we get integer number which is the identical as the integer number that is the outcome for using multiplication and addition on input Mi.

Decryption: M =C (mod p)

To verify accuracy of the scheme:

C = M r. p – r +1 mod S

Dec = C mod p

= M r. p – r +1 mod S mod p

= M r. p – r +1 mod p mod S

We recognize that a r. p – r +1 ≡ a (mod p) (Euler's theorem)

= M mod S = M, M< S (proved).

Algorithm 1: (SA)

1- Key Generation

- create two prime large integers p and q

S = p. q

- generate random big integer r

- Encryption

C =M r. p - r +1 mod S

- Decryption

M = C (mod p).

- Evaluation

Suppose there are two ciphertext:

C1 = M1 r1. p – r1 +1 mod S

C2 = M2 r2. p – r2 +1 mod S

C3 = C1+ C2 → M3 = Dec (C3) (or M3= M1+ M2)

C4 = C1. C2 → M4 = Dec (C4) (or M4= M1. M2)

Let C= f (C1... Ci), such as:

C= [(C1. C3) + C2] .C4

Let M= f (M1... Mi), such as:

M= [(M1. M3) + M2] .M4,

Where f is any function (Addition and Multiplication) applied on the messages or ciphertexts.

C (mod p) ≡ M, where M <p, except that, we must take (M mod p)

2.6.1.2.4.1 Time Complexity of SA scheme

Suppose n is the size of message in the type of decimal digits.

Encryption function:

C = M r * p – r +1 mod S

Then:

T (C) = O ((log (n))3).

Where, (log2 n) is the number of bits of n.

T (Encryption) = O((log (n))3) bit operation, according to the most costly operation.

Also,

T (Encryption) = O (log (n)) arithmetic operation.

Deciphing function:

M = C mod p

Then:

T (M) = O ((log (n))2)

T (Decryption) = O ((log (n))2) bit operation, according to the most costly operation.

Also,

T (Decryption) = O (1) arithmetic operation.

2.6.1.2.5 FHE in Cloud Security

Completely Homomorphic Encryption (FHE) schemes can solve the security issues of data stored in cloud. The information need to be encoded with FHE before sending it to the cloud. Next, the customer logins and make use of the server 's key generation to produce the private key, the customer is the sole holder of this private key. Now, the consumer encodes the information which you want to keep in the cloud.

The integrity and non-repudiation can be guaranteed during transmission by using other technologies such as digital signature to enhance safety. At the point when the client needs the server to perform a few calculations on this encoded information, (for example, search), he can send scrambled solicitation to the cloud server. The server plays out the necessary tasks and sent the scrambled outcome to client. At long last, the client unscrambles the information with his mystery key to recover the right outcome (Hemalatha & Manickachezian, 2014). Figure 5 outlines the way toward utilizing FHE to distributed computing.

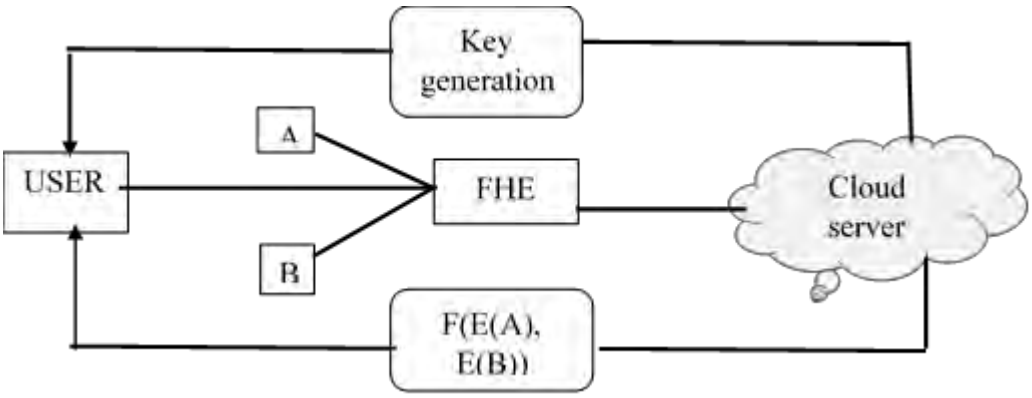


Figure 4: Application of total homomorphic encryption to secure the cloud (Ishan & Saad, 2016)

Encrypted data stored in cloud got an advantage of enhancing the measures for security of unreliable structures or software applications that stores or operates on delicate information. Cloud computing scenario is illustrated below.

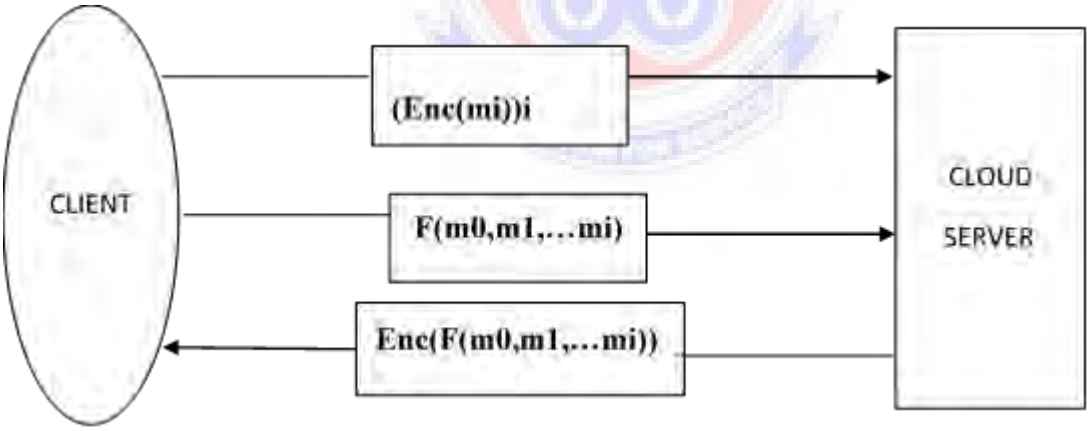


Figure 5: The Data Security Scheme for Cloud Computing (Yasmina & Rahal, 2015)

a) The public and secret keys or private keys are created by clients' application.

b) The public key is used to decipher data by application before sending it to the cloud server.

c) The user place a demand from the cloud to compute for the function f(m0,…,mi).

d) The request sent by the client is calculated for result by the cloud sever.

e) The cloud server calculates f(Enc(m0),…,Enc(mi)) without knowing {mi}i.

f) The outcome of the f(Enc(m0),…,Enc(mi)) is sent back to the user.

g) The private key Dec(f(Enc(m0),…,Enc(mi))) is used to decrypt the encrypted result and he customer can decipher the cipher outcome using the private key and get the same outcome as if the computation was done on raw data.

Example

Assuming company H has a very important data set that consists of the numbers 3 and 4. Company H encrypts the data so that 3 becomes 13 and 4 becomes 26. The company sends the coded set to cloud for safe keeping. Few months later, H needs to sum the numbers 3 and 4.

The encrypted data is then processed: the result 39 (13+26 = 39) can be downloaded from the cloud and company H can decrypt it to provide the final answer 7 (Yasmina & Rahal, 2015)

2.6.2 Execution time of the homomorphic encryption schemes

Table 1: The execution time of homomorphic encryption schemes, SAM, SDC and DGHV (Shihab & Ali, 2018)

Length of the message	**DGHV scheme (ms)**	**SDC scheme (ms)**	**SAM scheme (ms)**
2.8 k byte	6715148	42225899	72901
1.4 k byte	1241820	1283068	20818
12 byte	1118	1180	1007

Table 2: Execution Time of SA, Egamal and RSA Schemes (Shihab & Ali, 2018)

Size of message	32 bit ElGamal (m.s)	RSA (m.s)	SA (m.s)	64 bit ElGamal (m.s)	RSA (m.s)	SA (m.s)	128 bit ElGamal (m.s)	RSA (m.s)	SA (m.s)	256 bit ElGamal (m.s)	RSA (m.s)	SA (m.s)	512 bit ElGamal (m.s)	RSA (m.s)	SA (m.s)
12 Byte	15	16	24	15	17	20	21	22	27	22	24	45	35	23	73
1 K Byte	9491	11377	11581	15137	15625	15860	30751	36431	39041	34541	39965	74470	93711	37787	120354
1.5 K Byte	21497	25149	25278	33456	34589	35050	72440	90424	93189	82953	96749	173277	212789	93554	297891
2 K Byte	37816	45866	46186	58738	61477	64608	140054	163767	169649	155187	172906	325499	394844	178951	534314
2.5 K Byte	58699	72900	78159	100048	105859	107814	217615	263558	258449	241098	276159	488977	658823	287563	840937

2.6.3 Security of the Homomorphic Encryption Schemes

The DGHV plot is a completely homomorphic encryption plot over the numbers with its security dependent on the quality of the "rough whole number most prominent regular divisors" (GCD) issue made by Howgrave-Graham (2001). The DGHV plot demonstrate protection from a few distinct sorts of attacks to get the secret key even with Brute power attack within any event 2λ time. This is because of choosing the proper parameters for the plan. It is anyway additionally demonstrated that utilizing the cross section decrease calculation, the plan can be attack to get or recoup the plaintext structure of the ciphertext. DGHV plot considered the parameters settings which have been utilized in the assault to be suitable.

The RSA plot security depends on the point or truth that it isn't hard to duplicate two colossal primes to make a modulus the reverse activity of figuring the modulus into its prime variables can be troublesome. It is troublesome until care is taken on the whole number factorization issue for the spans of the numbers in question. The RSA cryptosystem security is based on the difficulty of factoring big integers. Endeavoring to break RSA by building up a number factorization answer for the moduli included is known as a numerical assault. That is, a numerical assault on RSA comprises of finding the prime components p and q of the modulus n. Obviously, knowing p and q, the aggressor will have the option to find the private exponent d for unscrambling. Another method of expressing equivalent to above would be that the aggressor would attempt to find the totient $\varphi(n)$ of the modulus n. In any case, as expressed prior, knowing $\varphi(n)$ is proportionate to knowing the variables p and q. On the off chance that an attacker can by one way or another make sense of $\varphi(n)$, the aggressor will have the option to set up the condition $(p-1)(q-1) = \varphi(n)$, that, alongside the condition $p \times q = n$, will permit the attacker to decide the qualities for p and q. Throughout the years, different numerical procedures have been produced

for taking care of the whole number factorization issue including enormous numbers, for example, Trial Division, Fermat's Factorization Method, Sieve Based Methods, and Pollard-ρ Method (Shihab & Ali, 2018).

ElGamal's security depends on the matter of the discrete logarithm. A discrete force is executed to encrypt and decode a message. An aggressor who needs to unscramble an encrypted message could try to get the private key. A logarithm should be registered to this end. There is no particular strategy for this despite certain preconditions on the underlying gathering. The encryption is secure under these conditions (Shihab & Ali, 2018).

ElGamal has the downside of making the ciphertext twice as long as the plaintext. It has the advantage that each time it's encrypted the same plaintext gives a different ciphertext. The ElGamal algorithm today is used in various cryptographic products. The encryption is secure under these conditions. The Elgamal cryptosystem is based on the difficulty of calculating the discrete logs in a huge prime modulus (Shihab & Ali, 2018).

According to Shihab and Ali (2018), the security of **SA** scheme depends on the right selection of a private key that must be unknown or uncommonly used, where knowing the secret part creates the chance of decoding easy. When the secret key is not known then it is highly impossible to decode the message in a limited period of time.

CHAPTER THREE

METHODOLOGY

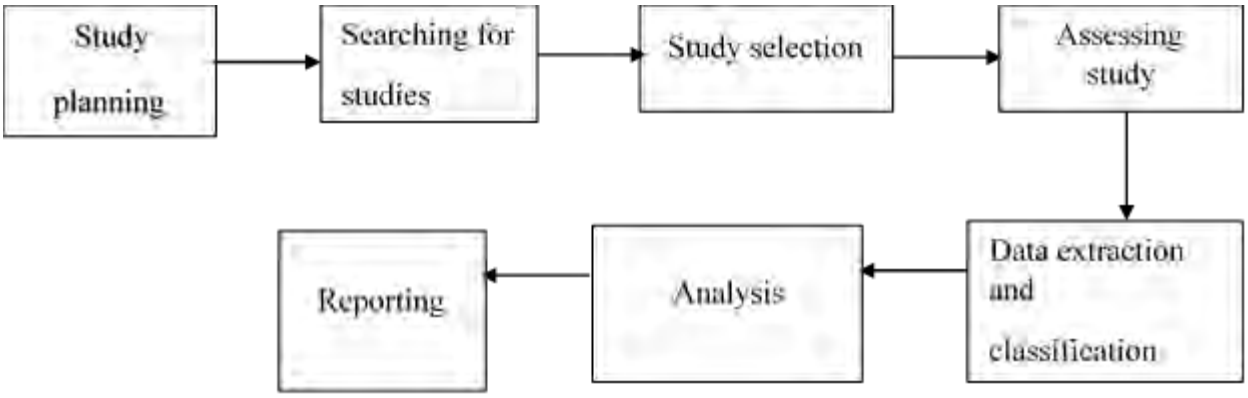
3.1 Introduction

This part of the study provides more explanation of the methodologies relevant to this study. This section of the research work outlines the means by which the data was taken and scrutinized, the strategies used for the study, justification of the choice of research, and finally ends with some limitations to the methodology used for the research.

3.2 Research Process

Research process comprises of the order of activities or steps essential to successfully carry out research and the preferred advancement of these steps (Shanti & Shashi, 2017).

Planning is one of the critical issue to consider when undertaking a SMS (Systematic Mapping Studies) because it is an activity that is time- consuming. In order to achieve a successful outcome, planning is the key element to consider. It also helps the researcher to be sure that, such SMS is needed and feasible. The Figure **6** represents a research process (i.e. study planning, searching for studies, study selection, assessing study quality, data extraction, data classification, analysis, and reporting).



Source: Researcher's Field Work, 2020

Figure 6: Flow chart of the research process

3.3 The research method

A comprehensive literature survey is conducted by the scholar to achieve the goal of the study. There are two main approaches to conducting literature studies. These, include Systematic Mapping Studies (SMS) and Systematic Literature Reviews (SLR) (Petersen, Feldt, Mujtaba, & Mattsson, 2008). If a researcher's intention is to identify, classify, and evaluate results corresponding to a specific research question then, SLR will be the best approach. On the other hand, if the researcher seeks to give responses to multiple research questions, then the SMS is the best option.

As the researcher aims to seek answers to multiple questions, it is appropriate to use the SMS approach.

Kitchenham and Chaters (2007), Petersen et al. (2008) define SMS as researches that are conducted with the intension to categorize and analyze studies conducted earlier on.

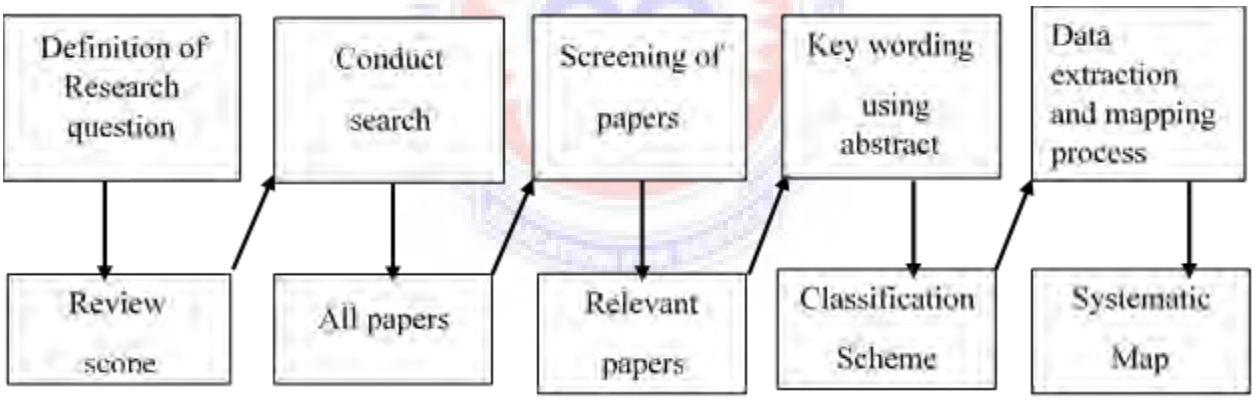
Such a study classifies and structures a field of interest in research by categorizing publications and analyzing their publication trends (Petersen et al., 2008). As far as this study is concerned

the researcher applies SMS approach and a procedure developed by Petersen et al. (2008) used in software engineering field.

Using systematic mapping as a mechanism, the researcher will be able to conduct a survey of valuable studies, obtaining relevant literature to be able to present credibility to the problem being investigated.

In order to find answers to the problem, the researcher outlines the stages of the systematic mapping studies (SMS) and procedure to be able to get the most needed earlier studies conducted relating to the topic. There are five steps that are involved in the search process in which the outcome from each step becomes the input for the next step as depicted in the figure 7 below:

Process Steps



Outcomes

Figure 7: systematic Mapping Process (Petersen, 2008)

From the figure 7, the steps are listed and described below

- Based on the objectives of the study, the research questions were defined. The systematic mapping studies have a main goal of giving the general view of the area of research and finds the number and kind of studies and outcomes presented in them.

- Conducting search with the help of predefined search queries based on the research questions.
- Searching of relevant articles on earlier studies with the help of search strings databases and scientific libraries. Earlier works done relating to the topic kept in databases and scientific libraries are found by the search strings. With the help of inclusion and exclusion criteria, all papers of the first group of studies are screened to identify the most relevant articles to be considered for the study. The above criteria defined, passes through series of stages beginning with the reading of the titles, abstracts and finally if possible full texts reading.
- Selecting all the relevant papers from the chosen set of articles by using keywording abstract. Here, in order to decrease the period of time required to develop classification pattern and making sure the pattern takes into account the current studies, then keywording is a major way to do it.
- Extracting data to answer the research questions. Having the classification scheme in place, the articles that are relevant are group sorted into the pattern. This means that the real data extraction takes place.

3.4 Rationale/Justification for Systematic Mapping Studies

From what have been explained by the researcher above, it is obvious that SMS is the best approach for conducting a relevant study. Systematic Mapping Studies is ideal for studies conducted in the fields where limited literature was reviewed relating to the topic and also in the situation where it is very needful of getting a broader view of your interest area, hence SMSs can

be applied to find out if there are any research gaps in the chosen area of interest (kitchenham & charters, 2007).

Systematic Mapping Studies approach is preferable where the researcher needs to scrutinize and categorize articles into groups from a larger dataset.

On this note, the systematic mapping studies was chosen by the researcher because of the following reasons:

- Search terms in the Systematic Mapping Studies (SMS) are less focused: The search terms for SMS are often less focused than SLR and likely to bring back huge figure of studies.
- Systematic Mapping Studies have broader research questions: Mapping studies largely have broader research questions leading them and often give answers to multiple research questions.
- Data extraction in Systematic Mapping Studies is broader: The process of fetching out data for mapping studies are similarly much broader than the process of fetching data for systematic reviews, and can accurately be termed as classifications or categorizations stage. The purpose for this stage is to categorize papers with sufficient details to give answers to the broad research questions.
- Systematic Mapping Studies gives a summary of analysis made: In the analysis stage data that is used to answer the research question posed is summarized. `

Nonetheless, one negative issue/flaw with the SMS is that, it is usually conducted by searching for data in the digital libraries and surveying academic databases. Thus, some important articles and publications are likely to be omitted by the researcher during the survey.

3.5 Search Strategy

This work contains a lot of literature scrutiny, explanation of other researchers' opinions concerning the topic and investigating all the answers to the research questions as a qualitative study.

Upon thorough consideration of the nature of the study, an intensive online survey was conducted to attain the goals of this research work. The researcher made good use of the internet to have access to online materials such as journals; articles etc. to enable him come out with a good work. The researcher formed Search Strings using keywords such as "cloud computing", "cloud security", "Homomorphic Encryption", "Application of Homomorphic Encryption", "Encryption Schemes". The researcher surfed the net manually and used search strings on scientific databases for relevant journals and articles. The researcher also adopted the "AND" Boolean operator to limit the search assortment process to publications pertinent to the objective of this dissertation.

Searches are conducted via digital libraries and databases which include, IEEE Xplore, Science Direct, Springer, and Google Scholar.

Also, the researcher customized each digital library based on their search rules to adapt to the general search strings. To keep on with important articles for my study, the researcher used "inclusion" and "exclusion" criteria.

3.5.1. Inclusion-Exclusion Criteria

In the digital libraries, the researcher filtered out the search results by applying inclusion and exclusion criteria. The core selection criteria are presented as follows:

- Selection by “title” and by “abstract”: the researcher first of all chose papers only with “cloud security”, “encryption schemes” and “homomorphic encryption schemes” terms in the titles or in the “abstracts”.
- Selection by full paper reading: papers written in English Language, and with research contribution in cloud and related to security and homomorphic encryption schemes are properly read and if possible selected. The relevant articles are extracted and categorized below.

3.5.1.1 Inclusion

- Articles with titles related to cryptosystem
- Articles having titles related to security in cloud
- Papers having titles similar to homomorphic encryption schemes
- Articles that compare homomorphic encryption schemes in cloud

3.5.1.2 Exclusion

- Articles from workshops
- Articles out of my study scope
- Articles that are not fully available
- Articles without Abstract

3.5.2 Execution Process

The search execution is done using automated search engines. During the screening process of relevant articles based on the inclusion-exclusion criteria defined previously, the researcher first examined titles and then, abstracts keywords. For those without sufficient details in their

abstracts, the researcher does full contents reading especially in papers written in English Language, and with research contribution to the research topic.

3.5.3 Conducting the Search

When the systematic mapping (SM) approach was implemented, all works concerning homomorphic encryption, cloud computing, security and privacy regarding data were found.

In this case, using search strings in IEEE Xplore, Network Digital Data Library of Thesis and Dissertation (NDLTD), Science Direct, ACM Digital Library and Springer Link, a total of 1003 journals and academic articles were retrieved.

The researcher applied the search string to each digital library and database and fetched important studies through the following steps:

Step 1: The researcher typed the search strings in the search engines of Google Scholar, UEW digital Library, Springer digital databases, IEEE Explore, and Science Direct, the researcher had 1003 papers.

Step 2: After Checking whether the study falls within the study scope, the researcher got 526 articles.

Step 3: The researcher obtained a total number of 241 papers after Checking if the study has full text on the web.

Step 4: The researcher checked if the article is written in English language and the researcher had 127 articles.

Step 5: After Checking if there are duplicates, the researcher obtained 84 articles.

Step 6: Check Title, abstract and keywords the researcher had 52 articles

Step 7: The researcher finally obtained 29 relevant papers for the study after applying the selection criteria.

Table 3: Description of the steps in Review

Study Type	Steps	Description
Primary	First Round (1st)	Typing the search strings in the search engines
	Second Round (2nd)	Check whether the study falls within my study scope
	Third Round (3rd)	Check if the study has full text on the web
	Fourth Round (4th)	Check if the study is written in English language
	Fifth Round (5th)	Check if there are duplicates
	Sixth Round (6th)	Check Title, abstract and keywords
Secondary	Seventh Round (7th)	Full text reading to check if it can be included in my study

Seven (7) stages were used to analyze the process for the "inclusion" and "exclusion" of studies the were gathered collected studies. At every stage, there was a thorough scrutiny of the studies collected to enable the researcher to sort out the desired and most important related to the study

at hand. The number of studies were reduced to fifty-two (52) potential studies for the initial selection. when six stage went through. Hence, to securely store any information in the cloud became an important concern in this era.

The final stage of selection commenced by implementing the last (seventh) round with the target of identifying studies that compares schemes with time complexity, space complexity and security of the protocols to securely store data in the cloud, immediately the outcomes of the initial selection accomplished.

The search outcomes and sorted articles from each digital library are displayed in Table 4 below.

Table 4: The search results and selected articles from each digital library

Database	**No. of Articles Obtained by Search Query**	**No. of selected Articles**
IEEE Xplore	357	11
ScienceDirect	217	7
Google Scholar	156	4
UEW Library	99	2
Springer	174	5
Total	**1003**	**29**

3.6 Data Analysis and Classification

It is very important for the data to be complete, accurate and suitable for future analysis (Sekaran & Bougie, 2010). For the analysis and classification of data, key wording grouping is most important in the systematic mapping process. To search for key terms and concepts about the research contributions, the researcher studied thoroughly the title of each paper, then followed by the abstract and full paper reading if it is necessary. When the researcher was reading the abstract and full paper he grouped important/relevant papers into research contribution to enable him map the selected studies according to their research focus and context. the researcher grouped the relevant papers into research contribution to help me to put up a cluster from the grouping/classification scheme. Thus, the relevant publications are clustered into groups as follows:

- Current Advance homomorphic encryption schemes category: These include Partial Homomorphic Encryption (RSA, Pallier and Elgamal) and Fully Homomorphic Encryption (DGHV, SDC, SA and SAM).
- Encryption schemes compared based on “execution time” category
- Encryption schemes compared based on “time complexity” category
- Encryption schemes compared based on “security” category

The researcher also identified other research type to add to those mentioned above.

- Philosophical papers: these papers provided advance ways of critically looking at a problem in terms of concepts and physical structure.
- Opinion papers: these papers present the author’s subjective view on a topic.
- Validation papers: these papers confirmed previous obtained results of other studies.

For clarity and easy understanding of the research findings, it is important to state clearly that, the search results are discussed based only on advance homomorphic encryption schemes, and other parameters such as execution time, time complexity and security to determine the best Homomorphic Encryption Schemes in respect to the purpose of this research.

Table 5: Research categories and number of articles

Research category	Number of articles for each category	percentage
Homomorphic encryption schemes	9	31.0
Execution Time	5	17.2
Time Complexity	4	13.8
Security Facets	8	27.6
Opinion Papers	1	3.4
Validation Papers	1	3.4
Philosophical papers	1	3.4
Total	29	100

3.7 Threats to Validity/Limitations

Validity is the means of using different techniques to check the accuracy of information collected from the field and literature (Creswell, 2009). Validity is to what degree a score actually reflects a definition. Simply put, it is the precision of the measuring device and represents a scale's ability to measure what it is intended to measure (Cooper & Emory, 1994; Zikmund, 2000).

According to Saunders, Lewis and Thornhill (2009), there are six fundamental validity challenges that are history, research, instrumentation, aging, maturation, and uncertainty about the course of causation.

The method used in this research may have some known threats to validity that can render the results of the study bias. As a result, those threats are given much attention and more efforts have been put in place to check the risks. The key threats which may occur when conducting SMS include the following:

Firstly, the selection criteria used, the inclusion and exclusion criteria only consider articles in a top-down approach; from titles, abstracts, and full contents reading and also, articles published only in English Language. In this case, there is the possibility that the researcher may perhaps exclude some important articles and Journals which are published in other languages. However, the researcher made good use of the language translators to curb the situation.

Again, there is the likelihood that the researcher did not make use of some libraries with relevant publications related to his study, as a result some relevant articles may not be captured. However, this risk is checked with the fact that most publications in the libraries and databases selected contains items of the same kind, hence article redundancy decreases this risk.

Also, there is a possible risk that, the search engines might have made used of other irrelevant publications because the search criteria of the study are defined on the assumption that, the work should only be oriented towards publications related to my research questions. The researcher employed the use clear terms with the "AND" logical operator to make search strings to mitigate this risk.

Finally, the researcher also, chose digital libraries and databases depending on their ranking in scientific research which the researcher consider should contain articles relevant to the study.

Due to this, there is the likelihood that the researcher may not integrate some libraries containing relevant information related to my topic. So, some important articles may be left out.

3.8 Conclusion

In this chapter, the methodological approaches available for the study are explained. The chapter also highlights how the data for the research is gathered and scrutinized/analyzed, including the research strategies, justification/Rationale of the research choice, and finally concluded with the main limitations to the methodological approaches used by the researcher.

CHAPTER FOUR

DISCUSSION OF FINDINGS

4.1 Introduction

From the previous sections, data security and privacy issues that cause hindrances for storing data securely in the cloud have become an issue of concern to both clients and cloud service providers in the cloud environment.

4.2 Discussion of Parameters

Even though privacy and security of data are key issues concerning cloud computing, technology already revolutionized to a level that many security measures are designed to protect user data stored in cloud. As such, research should be focused more on introducing the homomorphic encryption schemes appropriate to provide security and privacy to data store in cloud. This chapter thus, deliberates on the findings of this research work based on the research questions.

Table 6: Size of message, execution time, time complexity and security of homomorphic encryption schemes

Homomorphic Encryption Schemes	Size of Message	Execution Time (ms)	Time Complexity	Security of the Protocol
SA	12 Byte	24	$O((\log (n))^3)$	Depends on the right selection of a private key that must be unknown or uncommonly used
SAM	12 Byte	1007	O((log (n))	Depends on selecting a prime

			^{2})	number as a secret key
SDC	12 Byte	1180	O (n^2)	The method (process) of deciphering depends on the secret (private) key **p** and retrieval method (process) depends on the integer **q**.
DGHV	12 Byte	1118	O (n^2)	Depends on the powerful nature of the approximate integer GCD)
RSA	12 Byte	16	O ((log (n))^3)	Relying on the toughness of factoring huge integers
Elgamal	12 Byte	15	O ((log (n))^3)	Relying on the toughness of calculating discrete logs in a huge prime modulus

4.3 Research Question one (1): What are the current advance homomorphic encryption schemes available in the cloud?

This research question basically has been answered in Chapter **2.** A number of advance Homomorphic Encryption Schemes used to store data in the cloud were identified. Some frameworks and algorithms for the various homomorphic schemes for storing data securely in the cloud were presented to clarify the advance homomorphic encryption schemes as compared to other traditional encryption schemes. Thus identifying a number of Homomorphic Encryption Schemes used to store data securely in cloud.

Some of these encryption schemes identified in the literature review include, partial homomorphic encryption schemes (RSA, Elgamal) and Totally homomorphic encryption schemes (SA, DGHV, SDC, SAM).

The literature survey identified **1003** published articles and academic journals of which **29** of these publications were relevant as the study is concerned since they demonstrated how the homomorphic encryption schemes can be used to securely store data in cloud, as such, enough to be considered for review. It can also be realized in table **5** that out of **29** published papers, **9** representing **31.0%** of the relevant papers published talked about current advance homomorphic encryption schemes available in the cloud.

During the Systematic process, that, all relevant papers discuss measures to secure end-to-end communication. Generally, it has been recognized that the security of homomorphic encryption schemes was well discussed by most papers but the algorithms are not given in details to alleviate public concerns about the encryption schemes.

With respect to privacy and security of data in cloud computing, a reasonable number of researches have been conducted on it. However, the focus is mainly on encryption schemes that the cloud service provider stands the chance of sharing the encryption/secret key with the client. Sharing the key with the service provider may sometimes be dangerous and can render the data insecure. As such more research into this area where cloud service provider does not have to share the secret/encryption key with the user is required.

Other encryption schemes available to curb privacy and data security challenges in cloud environment include Elliptic Curve Cryptography, Searchable symmetric, public key with keyword search and Deterministic encryption.

4.4 Research Question two (2): Which advance Homomorphic Encryption Schemes is the best for storing data securely in cloud?

Considering the security issues in the cloud environment, many scholars and researchers have published papers and academic journals containing different security issues and ways of protecting users' data that is stored in the cloud. A good number of the studies that look into cloud computing architectures define the layers' architectures. Most commonly, they divide the cloud storage architecture into five distinct layers; multiple/many users, internet/network front end, logic control and back end emphasizing how delivery of storage on request in an extremely climbable and multi-tenant way as shown in figure **1** in the previous section. The researcher has also identified that, previous studies carried out on cloud data storage by other researchers reported some Homomorphic Encryption Schemes that are used to store data securely in cloud as; fully homomorphic schemes (DGHV, SDC, SAM and SA) and partial homomorphic schemes (RSA, Elgamal and pallier) as shown in figure **3.**

Considering table **6** derived from homomorphic encryption by prime modular operation and homomorphic encryption based on Euler's theorem, the message size of **12** bytes can be executed within 24ms by the SA encryption scheme as the minimum execution time and Elgamal, RSA, SAM, DGHV, SDC with execution times as 15ms,16ms,1007ms, 1118ms,1180ms respectively. Elgamal with 15ms and RSA with 16ms as execution times are not considered by the researcher because they are partial homomorphic encryption schemes.

Again, considering chapter **2** and table **6**, the homomorphic encryptions schemes such as SA, Elgamal, RSA, SAM, DGHV, SDC have their time complexities as O ((log (n))3), O ((log (n))3), O ((log (n))3), O ((log (n)) 2), O (n2) and O (n2) respectively.

Also, considering the security level of the following homomorphic encryption from Homomorphic Encryption by prime modular operation, it can be realized that, the DGHV is a completely homomorphic encryption plot over the numbers with its security dependent on the quality of the "rough whole number most Greater Common Divisors" (GCD). The DGHV plot demonstrates protection from a few distinct sorts of assaults to get the mystery key even with Brute power assault within any event 2λ time. This is because of choosing the proper parameters for the plan. It is anyway additionally demonstrated that utilizing the cross section decrease calculation, the plan can be assaulted to get or recoup the plaintext structure of the ciphertext.

The **SDC** scheme, attacker cannot easily obtain the plaintext although the attackers may be able to get the large integer q, by means of c mod q, the result they can get is only the m+ p, but the plaintext m cannot leak out.

Choosing the prime number as a secret key in the **SAM** scheme improves the security than when selecting any number. The motive behind this is that, the third party cannot easily get the secret key by choosing randomly. Therefore, it is costly for attackers, because even if they get the number, it has to be verified and confirmed is prime and tested on the encryption equation. Going through these processes consumes a lot of time and requires several attempts to break through the code. For this reason, the prime number give the robustness to SAM security and one probability of the solution.

Also, any time the customer wishes to return the encrypted data store on the cloud server with the cipher retrieval algorithm, the SAM scheme always directs a constant big integer q to the cloud server.

Again looking at the homomorphic encryption based on Euler's theorem, the **RSA** security depends on the fact that it is easy to multiply two large primes to construct a modulus, the

inverse operation of factoring the modulus into its prime factors can be very difficult, the difficult until the integer factorization problem is solved for the sizes of the numbers involved. Attempting to break RSA by developing an integer factorization solution for the moduli involved is known as a mathematical attack. That is, a mathematical attack on RSA consists of discovering the prime factors p and q of the modulus n. Clearly, knowing p and q, the attacker will be able to discover the private exponent d for decryption. The safety of RSA cryptosystem depends on the difficulty of factoring big integers.

ElGamal's protection depends on the question of a discrete logarithm. A discrete force is exercised to encode and decode a message. An aggressor who needs to unscramble a message may try to get the private key back. There is no particular strategy for this despite any preconditions on the underlying gathering. Within such conditions the encryption is secured. ElGamal has the downside since it makes the ciphertext twice as long as the plaintext. It has the advantage that a different ciphertext is provided each time it is encrypted in the same plaintext The Elgamal algorithm today is used in various cryptographic products. The safety of ElGamal cryptosystem is based on the difficulty of calculating discrete logs in a large prime modulus.

The security of **SA** scheme depends on the right selection of a private key that must be unknown or uncommonly used, where knowing the secret part creates the chance of decoding easy. When the secret key is not known then it is highly impossible to decode the message in a limited period of time. Also depicted in Table **5** are the results of various research categories that have been conducted on cloud data storage. The results show that, as much as **31.0%** of the studies reported the advance homomorphic encryption schemes that are used for cloud data storage, **17.2%** of the survey papers are homomorphic encryption indicating "execution time", **13.8%** of the studies

reported about homomorphic encryption with parameter “time complexity” and **27.6%** of studies reported homomorphic encryption with parameter “security”.

Other publications identify by the researcher include; opinion papers, which represent **3.4%,** validation papers are **3.4%,** and **3.4%** are identified as philosophical papers.

CHAPTER FIVE

SUMMARY, CONCLUSION, RECOMMENDATION AND FUTURE WORK

5.1 Introduction

This chapter presents a summary of the research findings, conclusions, and recommendations for future research.

5.2. Summary of Findings

This study aims to investigate the advance homomorphic encryption schemes in order to store data securely in the cloud. The researcher used a systematic mapping study approach because this method is appropriate for the purpose of a study since this approach is acceptable where the purpose of a study is to provide answers to multiple research questions. Study results shows that; the cloud world consists of a number of users who store and retrieve data in the cloud. The cloud service company provides the administration / management and infrastructure. So, once there's a lot of cloud users, data security and privacy are extremely important since masses of users have kept their vital information on these clouds, which is what makes it riskier to secure every bit of information. Data security has become a major problem in cloud computing, and if protection and privacy are ignored in cloud computing, then each user's private information is at risk, enabling easy cyber breaches to hack into the network and access private data stored by any user. SAM scheme demonstrates that it has a very good time complexity, it’s execution time is very fast and security can be trusted because its secret key is represented by a prime number and it is secure when retrieving ciphertext. Therefore, is the best Homomorphic Encryption scheme for storing data securely in the cloud.

Furthermore, some various advance homomorphic encryption schemes with their algorithms were presented, frameworks and cloud architecture that have been initiated by different scholars are highlighted. Out of the **29** related articles and academic journals survey by the researcher, a total of **31.0%** of the studies reported the advance homomorphic encryption schemes that are used for cloud data storage.

It is obvious that, cloud users have much concerns regarding data security and privacy and any data breach could undermine consumers' trust in the cloud data storage.

Finally, the findings of this dissertation provide a deep understanding of past research regarding the issues of privacy as well as security of cloud data storage and expose many advanced homomorphic encryption schemes that can be used to store data securely in the cloud. While the cloud data storage technology keeps advancing over years, confidence in, and acceptance of this technology will depend on the security and privacy of users' data stored in the cloud.

5.3 Conclusion

Cloud data storage is a technology that involves multiple users, devices, systems and services. Although the cloud data storage technology creates huge opportunities and benefits for the society, its data security and privacy measures are not good enough to keep up with the cyber breaches by intruders/hackers, as such creating great data insecurity and privacy risks.

The researcher knows what it means to store data in the cloud and must be well protected and secured with the help of encryption schemes that are fully homomorphic and also presented in details the categories of homomorphic encryption schemes with examples.

So far as this research is concern, the SAM scheme which is a very good and effective scheme that takes characters directly the way they appeared on the text and enter them into the encryption equation without converting the character which is in plain text to binary format. For this reason, in terms of execution time and time complexity, SAM scheme is very fast when comparing it with other encryption schemes considered in this study. SAM scheme demonstrates that its security can be trusted since it uses prime number as a secret key and it is very secure in cipher text retrieval. Therefore, SAM showed a good security for the stored data on the cloud.

Even though, many researches have been conducted in the area of cloud data storage security and how data is secured, it is obvious that, there is still apparent lack of research and literature that define completely the best encryption scheme that is homomorphic for storing cloud data.

It is also essential that cloud developers, service providers and users who used the cloud devices, services and systems work together with the software designers, programmers and users to come out with best encryption schemes to ensure that cloud data storage system is built on a foundation that is secured and trustworthy.

5.4. Recommendations

In this section, the researcher outlines some guidelines that can be practiced by cloud developers, service providers/management and users, to guarantee the needed data privacy and security of information stored in the cloud.

To ensure cloud data privacy and security in the cloud environment, developers and manufacturers should always consider the appropriate encryption schemes to handle security problems at the design phase of data storage devices and service systems development.

Considering security at the design phase of cloud development reduces possible risks and prevent complications and costs involve in trying to enhance the security of products when they were already developed and put to use.

Also, based on the complex nature and the advancements of cloud data storage development, it is quite vital to know the various advance homomorphic encryption scheme and adopt the best scheme to securely store data in the cloud to ensure quality accessibility of services.

Furthermore, security updates and vulnerability management are crucial factors to be considered in the cloud environment. Security updates and vulnerability management in software can be done through automated updates mechanisms, such that patches would be applied automatically.

More importantly, designers and developers of cloud data storage products should build devices that can readily recognized security problems in the cloud system. As this can help in identifying vulnerabilities, detecting abnormalities, responding to potential breaches, and recovering from damages done to cloud data storage systems.

It is obvious that, the cloud data storage systems consist of complex security problems which need to be addressed. It is my hope that if cloud data storage designers, developers and service providers pay attention to the above-mentioned recommendations, the future cloud data storage environment will not only provide good accessibility but will also be built to render a secured service that will store users' data safely.

5.5 Future Work

In this section, the researcher outlined some salient areas where further research would be conducted, based on the findings revealed while conducting this thesis.

First and foremost, cloud data storage technologies are still emerging, and while considerable amount of technical research is being conducted on the data security and privacy issues of the new technology, there have been very little research on the end users' opinions and qualitative analysis on this area. Therefore, it is recommended that, more qualitative research and analyses be done to understand the users' perception about the advance homomorphic encryption schemes for data security and privacy.

Future work will focus on developing and implementing more advance encryption schemes in cloud computing.

Again, the research was conducted using Systematic Mapping Studies, which is limited as compare to other approaches such as Systematic Literature Studies. Hence, further research can be carried out using different strategies such that, additional keywords can be included to prolong the survey in order to present a more complete and holistic coverage of the data security issues in the cloud.

REFERENCES

Aderemi, A., Atayero, & Oluwasey, F. (2011). Security Issues in Cloud Computing: The Potentials of Homomorphic Encryption. *Journal of Emerging Trends in Computing and Information Sciences*, *6*, 56-67.

Armbrust, M. et al (2009). *Above the clouds: A Berkeley view of cloud computing*. UC Berkeley Technical Report. No. UCB/EECS-2009-28 http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

Arockiam, L. & Monikandan, S. (2013). Data Security and Privacy in Cloud Storage using Hybrid Symmetric Encryption Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering,* 2, 8.

Arokia, R., Paul, R., S., & Shanmugapriyaa (2012). Evolution of Cloud Storage as Cloud Computing Infrastructure Service. *IOSR Journal of Computer Engineering (IOSRJCE)* 1, (1), 38-45

Ayub, H., M., Manish, R. & Monjul, S. (2015). A Brief Overview of Homomorphic Cryptosystem and Their Applications. *International journal of computer applications*, (*0975* – 8887) NCIT.

Baohua, C. & Na, Z. (2014). *Fully Homomorphic Encryption Application in Cloud Computing*, in Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 11th International Computer Conference.

Bikram, B. (2009). Safe on the Cloud. *A Perspective into the Security Concerns of Cloud Computing, 4*, 34–35.

Boneh, D., Goh, E. & Nissim, K. (2005). *Evaluating 2-DNF formulas on ciphertext*, in Proceedings of Theory of Cryptography, TCC'05, pp 325-341. Springer Berlin Heidelberg.

Boss, G., Malladi, P., Quan, D., et al. (2007). IBM Cloud Computing White Book, 1http://www-01.ibm.com/software/cn/Tivoli/ao/reg.html

Buyya, R., Yeo, C., Venugopal, S., Broberg, J. & Brandic, I. (2009). Cloud computing and emerging platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Comput Syst, 25*(6), 599–616.

Chunye, G., Jie, L., Qiang, Z., Haitao, C. & Zhenghu G. (2010). The Characteristics of Cloud Computing *2010 39th International Conference on Parallel Processing Workshops,* Pages 275–279 https://doi.org/10.1109/ICPPW.2010

Cooper, D. R., & Emory, C. W. (1994). *Business Research Methods* (5th ed.). USA: Richard D. Irwin.

Creswell, John, W. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods*. New York: Teachers College Press.

Dan, B., Goh, E., & Kobbi, N. (2005). *Evaluating 2-DNF formulas on ciphertext.* In Theory of Cryptography Conference, TCC, volume 3378 of Lecture Notes in Computer Science, pages 325-341. Springer, 2005.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory, 22.6* (1976), 644-654.

El Gamal, T. (1985). *A public key cryptosystem and a signature scheme based on discrete logarithms*. In Advances in Cryptology, pp. 10-18. Springer Berlin Heidelberg.

Eyad, S. (2015). *Processing Over Encrypted Data: Between Theory and Practice*, Proceedings of the 8th Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering.

Gentry, C. (2009). *A fully homomorphic encryption scheme*: *Doctoral dissertation*, Stanford University.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *STOC*. Vol. 9.

Gu Chunsheng (2012). *Attack on Fully Homomorphic Encryption over the Integers*, Cryptology ePrint Archive, Report 2012/157.

Hemalatha, S., & Manickachezian, R. (2014). Performance NHJBN of Ring Based Fully Homomorphic Encryption for securing data in Cloud Computing. *International Journal of Advanced Research in Computer and Communication Engineering*, Vol.3, Issue 11.

Howgrave-Graham, N. (2001). *Approximate integer common divisors*, in CaLC, pp. 51–66.

Ihsan, J., & saad N. A. (2016). Using Fully Homomorphic Encryption to Secure Cloud Computing. *Internet of Things and Cloud Computing*, *4*(2), 13-18. doi: 10.11648/j.iotcc.20160402.12

Iswarya, K. (2014). Security Issues Associated with Big Data in Cloud Computing, *SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE)*, 1(8), 1-8.

Jamil, D., & Zaki, H. (2011). Cloud Computing Security. *International Journal of Engineering Science and Technology, 3*(4), 3478–3483.

Jaydip, S. (2013). *Homomorphic encryption: theory & Application*, In Tech, Theory and Practice of Cryptography and Network Security Protocols and Technologies.

Kamal I., H., Kartit, M. A., & El Marraki (2014). Foremost Security Apprehensions in Cloud Computing *Journal of Theoretical and Applied Information Technology,* Vol. 59 No.3

Kerckhoffs, A. (1883). Military Cryptography (part i). *Journal of Military Sciences*, 9(1), 5–38.

Kerckhoffs, A. (1883). Military Cryptography (part ii). *Journal of Military Sciences*, *9*(2), 161–191.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. Technical Report EBSE 2007–001, Keele University and Durham University Joint, Report

Kokila, S. & Princess, T., R. (2015). Software as a Service, a Detailed Study on Challenges and Security Threats. *SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE),* 2(12), 9-14.

Kuyoro S. O., Ibikunle F., & Awodele O. (2011). Cloud Computing Security Issues and Challenges, *International Journal of Computer Networks (IJCN), 3,* 5.

Lauter, K., Naehrig, M. & Vaikuntanathan, V. (2011). *Can Homomorphic Encryption be Practical?* CCSW' 11, Chicago, llinois, USA, pp. 113–124.

Li, J., Song, D., Chen, S., & Lu, X. (2012). A Simple Fully Homomorphic Encryption Scheme Available in Cloud Computing, In Proceeding of IEEE.

Melchor, C., A. et al. (2011). *Improving Additive and Multiplicative Homomorphic Encryption Schemes Based on Worst-Case Hardness Assumptions*. IACR Cryptology ePrint Archive 2011: 607.

Mell P., & Grance, T. (2011). *The NIST Definition of Cloud Computing, National Institute of Standards and Technology*, U. S. Department of Commerce.

Menezes A., Van Orschot, P., &Vanstone S. (1997). *Handbook of Applied Cryptography*, CRC Press.

Nitin, J., Saibal, K., Pal, Dhananjay, K., & Upadhyay (2012). Implementation and Analysis Of Homomorphic Encryption Schemes. *International Journal on Cryptography and Information Security(IJCIS), 2, 2.* June 2012. DOI:10.5121/ijcis.2012.2203 27

Paillier, P. (1999). *Public-key cryptosystems based on composite degree residuosity classes*. In Advances in cryptology—EUROCRYPT '99, pp. 223-238. Springer Berlin Heidelberg.

Payal,V. P., Shraddha, B. P., Shafika, N. P., Niyatee, I. B., & Rutvij, H. J. (2014). Survey of Various Homomorphic Encryption Algorithms and Schemes. *International Journal of Computer Applications (0975-8887*), *91*(8).

Petersen K., Feldt R., Mujtaba S., & Mattsson M. (2008). *Systematic mapping studies in software engineering. In: Evaluation and Assessment in Software Engineering (EASE'08)*. Italy. University of Bari.

Rivest, R., Adleman, I., Dertouzos, M. (1978). On Data Banks and Privacy Homomorphism, *Foundations of Secure Computation, 4*(11), 169-180.

Saunders, M., Lewis P., & Thornhill, A. (2009). Research Methods for Business Students (5th ed.). New Jersey: Prentice Hall.

Sean, C., & Kevin C. (2012). Cloud Computing Technologies. *International Journal of Cloud Computing and Services Science (IJ-CLOSER) 1,* (2), 59-65.

Sekaran, U., & Bougie, R. (2010). *Research Methods for Business: A Skill Building Approach (5th edition)*. New Jersey: John Wiley and Sons.

Shanti, B. M., & Shashi, A. (2017). *Handbook of Research Methodology*: *A Compendium for Scholars & Researchers (Based on revised syllabus of research methodology of various universities)*. Educreation Publishing, India.

Shihab, S., H. & Ali, M., S. (2018). Design of Fully Homomorphic Encryption by Prime Modular Operation. *Telfor journal*, *10*(2), 34-56.

SO, K. (2011). Cloud computing security issues and challenges. *International Journal of Computer Networks*, *3*, 5.

Spoorthy, V., Mamatha, M., & Santhosh, K. B. (2014). A Survey on Data Storage and Security in Cloud Computing. *International Journal of Computer Science and Mobile Computing, 3*(6), 306 – 313 www.ijcsmc.com

Syam K. P., & Subramanian, R. (2011). An Efficient and Secure Protocol for Ensuring Data Storage: Security in Cloud Computing. *International Journal of Computer Science Issues, 8*(6), 1694-0814. www.IJCSI.org

Tebaa, M., Saïd, E., & Abdellatif, E. (2012). *Homomorphic encryption applied to the cloud computing security*. In Proceedings of the *World Congress on Engineering, 1*, 4-6.

Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010). *Fully homomorphic encryption over the Integers*, in Proceedings of Advances in Cryptology, EUROCRYPT'10, pages 24–43.https://eprint.iacr.org/2009/616.pdf

Van Tilborg, H., Ed. (2005). *Encyclopedia of Cryptography and Security*, Springer, New York, NY, USA.

Xiaowei, Y., Xiaosong, Z., Ting, C., Hongtian, Z., & Xiaoshan, L. (2012). The Research and Design of Cloud Computing Security Framework. DOI: 10.1007/978-3-642-25541-0_95

Xing G., Chen, X., Zhu, P., & Ma, J. (2006). A method of Homomorphic Encryption, *Wuhan University Journal of Natural Sciences*, *11*(1), 181-184.

Yang, J., Mingyu, F., Guangwei, W., & Zhiyin K. (2014) Simulation Study Based on Somewhat Homomorphic Encryption. *Journal of Computer and Communications, 2*, 109.

Yasmina, B., & Rahal, R. (2015). Secure Data in Cloud Computing Using Homomorphic Encryption. *Journal of Theoretical and Applied Information Technology,* 82(2). www.jatit.org

Zaid, K., Ali, A., Kamal I., H., El Marraki, M., M. Hedabou, Belkasmi, M., & Kartit, A. (2016). *Applying Encryption Algorithm for Data Security in Cloud Storage*. DOI: 10.1007/978-981-287-990-5_12

Zhang, S., & Chen, X. (2010). *Cloud Computing Research and Development Trend*. *In Second International Conference on Future Networks*, p. 93. IEEE Computer Society 1730 Massachusetts Ave., NW Washington, DC, United States.

Zikmund, W. G. (2000). *Business Research Methods* (6th ed.). USA: Harcourt.